Data-driven Markov Chain Monte Carlo

DDMCMC and its applications in
image segmentation and object recognition

A Bayesian Formulation

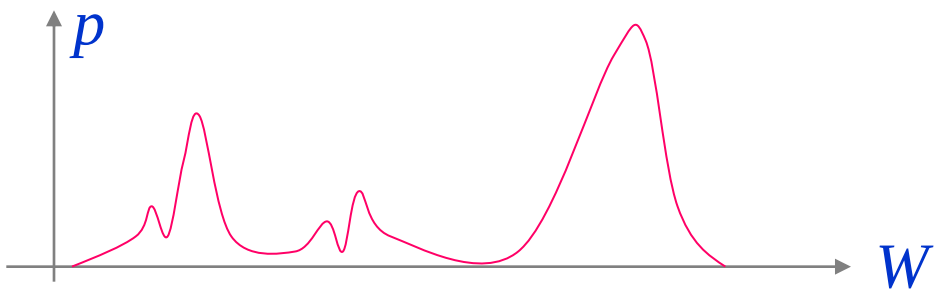
A basic assumption, dated back to Helmholtz (1860), is that biologic and machine vision is to compute the most probable interpretation(s) from input images.

Let I be an image and W be a semantic representation of the world.

$$\mathrm{W}^{*} = \arg\max_{\mathrm{w}\in\Omega} p(\mathrm{W}\,|\,\mathrm{I}) = \arg\max_{\mathrm{w}\in\Omega} p(\mathrm{I}\,|\,\mathrm{W})\,p(\mathrm{W})$$

In statistics, we sample from a posterior probability to preserve ambiguities.

$$(\mathrm{W}_1, \mathrm{W}_2, \ldots, \mathrm{W}_k) \sim p(\mathrm{W}\,|\,\mathrm{I})$$



What is Data-Driven Markov Chain Monte Carlo?

The complexity of sampling the posterior $\mathrm{W} \sim p(\mathrm{W} \mid \mathrm{I})$ is in the Metropolis-Hastings jumps

Consider a reversible jump $W_A \leftrightarrow W_B$

$$\alpha(W_A \to W_B) = \min(1, \frac{p(W_B \mid \mathrm{I})\, G(W_B \to W_A)}{p(W_A \mid \mathrm{I})\, G(W_A \to W_B)}) \quad \text{or} \quad \min(1, \frac{p(W_B \mid \mathrm{I})\, q(W_A \mid W_B)}{p(W_A \mid \mathrm{I})\, q(W_B \mid W_A)})$$

Without looking at the data, the pre-designed proposal probabilities are often uniform distributions, thus it is a blind (exhaustive) search !

In DDMCMC,

$$\alpha(W_A \to W_B) = \min(1, \frac{p(W_B \mid \mathrm{I})\, q(W_A \mid W_B, \mathrm{I})}{p(W_A \mid \mathrm{I})\, q(W_B \mid W_A, \mathrm{I})})$$

If $q(W_B \mid W_A, \mathrm{I}) \cong p(W_A \mid \mathrm{I}), \quad q(W_A \mid W_B, \mathrm{I}) \cong p(W_B \mid \mathrm{I})$

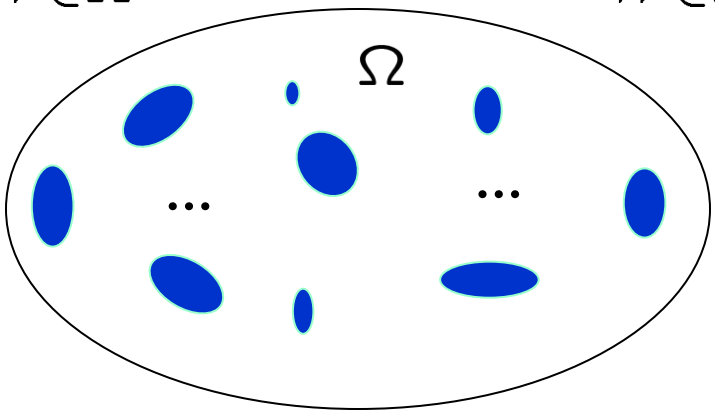
Then it may converges in a small number of steps !

An example

输入图像

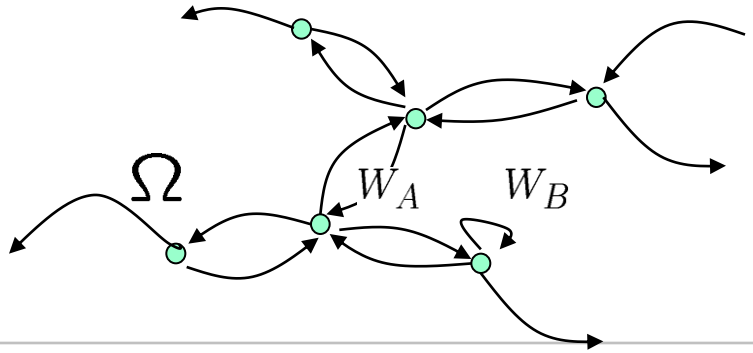
Sampling the Posterior Distribution

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



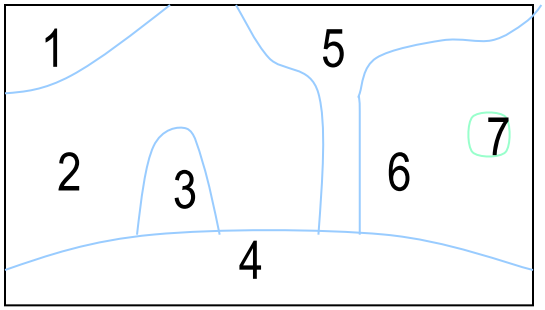
Markov Chain: $MC = (p,\ K,\ p_o)$

To design transition kernel: $p \cdot K = p \quad p_o \cdot K^n \to p$



Example: Image Segmentation

$$W = (n,\ \{R_i,\ l_i, \theta_i : i = 1, 2, \ldots, n\}) \in \Omega$$



$\pi_7 = (R_1, R_2, ..., R_7)$ is a 7-partition of the lattice.

$$\Omega_{\pi_n} = \{\pi_n = (R_1, R_2, ..., R_n) : \bigcup_{i=1}^{n} R_i = \Lambda, R_i \cap R_j = \varnothing, \forall i \neq j\} \ / \ pg$$

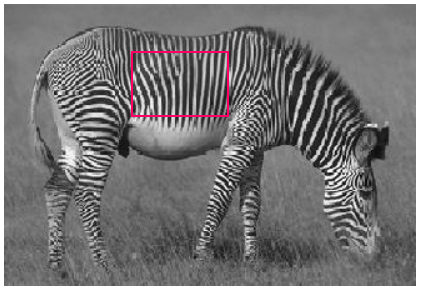
A permutation group

The *partition space* is $\Omega_\pi = \bigcup_{n=1}^{|\Lambda|} \Omega_{\pi_n}$

Likelihood Models

1 2 3 4






1 ■ : iid Gaussian for pixel intensities

$$p(\mathbf{I}_R; l_1, \theta) = \prod_{v \in R} G(\mathbf{I}_v - \mu; \sigma^2)$$

2 ■ : non-parametric histograms

$$p(\mathbf{I}_R; l_2, \theta) = \prod_{v \in R} h(\mathbf{I}_v)$$

3 ■ : Markov random fields for texture

$$p(\mathbf{I}_R; l_3, \theta) = \prod_{v \in R} p(\mathbf{I}_v | \mathbf{I}_{\partial v}; \theta)$$

$$= \prod_{v \in R} \frac{1}{Z_v} exp\{- < \theta, h(\mathbf{I}_v | \mathbf{I}_{\partial v}) >\}$$

4 ■ : spline model for lighting variations

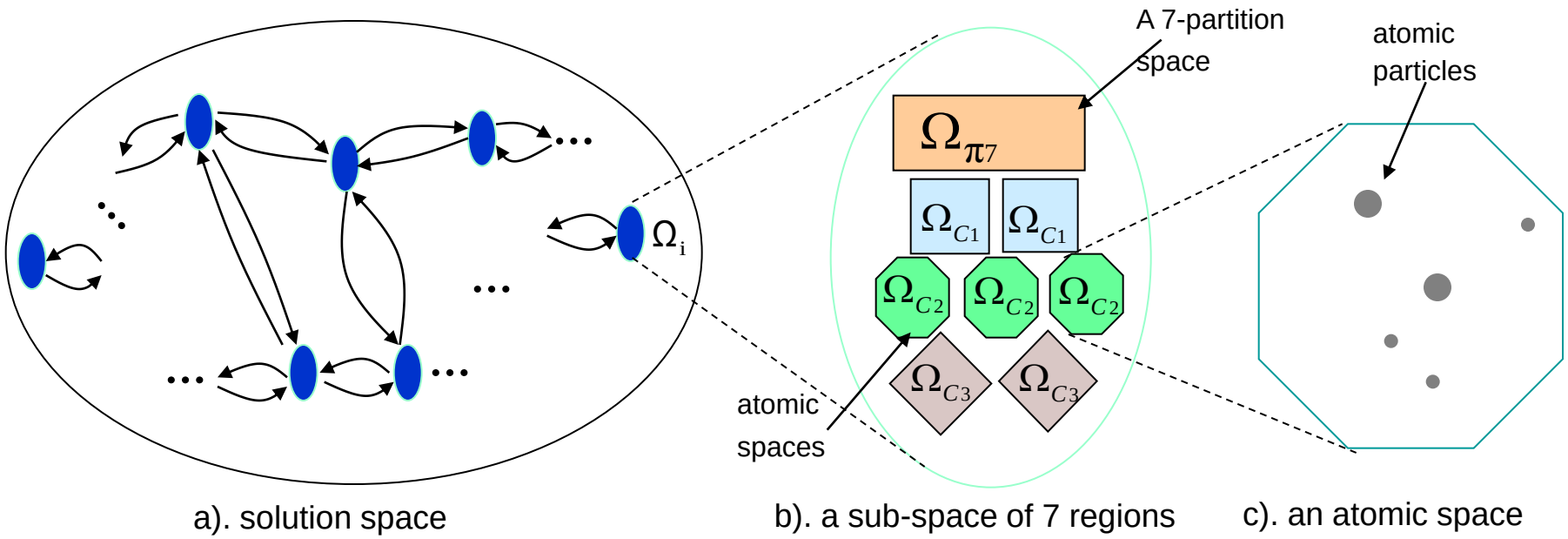
$$p(\mathbf{I}_R; l_4, \theta) = \prod_{v \in R} G(\mathbf{I}_v - B_v; \sigma^2)$$

5 ■ : iid Gaussian for color (LUV)

6 ■ : mixture of Gaussians for color

7 ■ : spline model for smooth color variations (e.g. sky, shading, …)

The Search Space

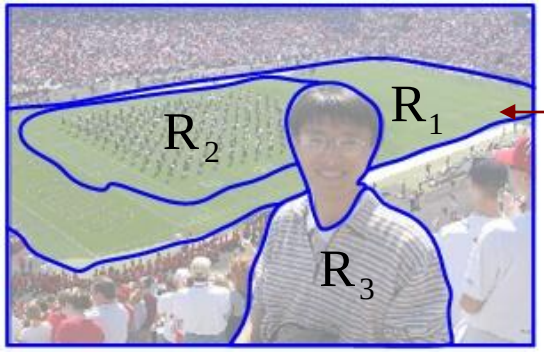


a). solution space

b). a sub-space of 7 regions

c). an atomic space

A Case Study: Image Segmentation by DDMCMC



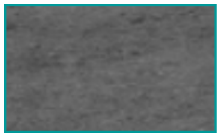
$$W = (n, \{(R_i, l_i, \theta_i), i = 1, ..., n\})$$

R_i: partition (by boundaries)

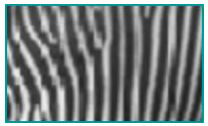
l_i: intensity type θ_i: parameters

grey-scale

color


flat

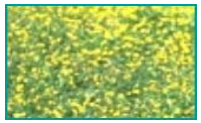

clutter

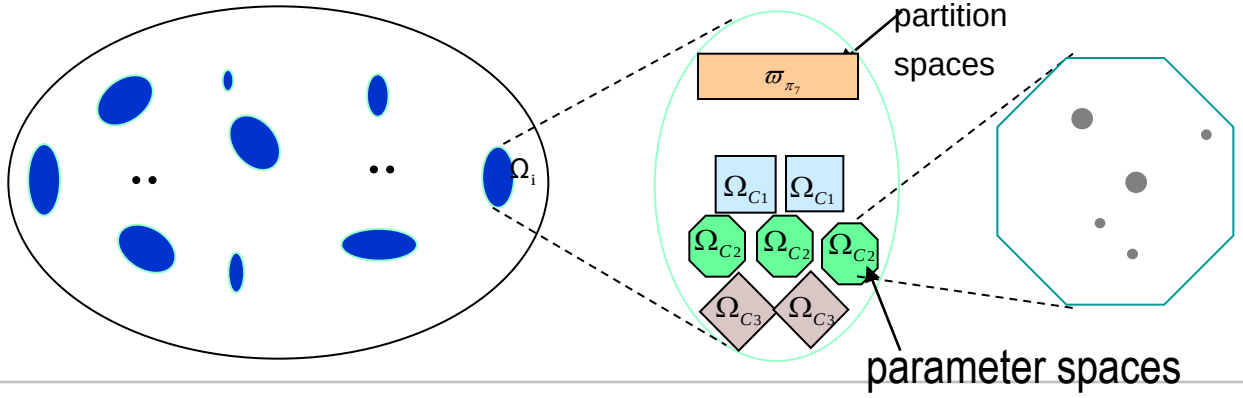

texture


shading


flat


shading

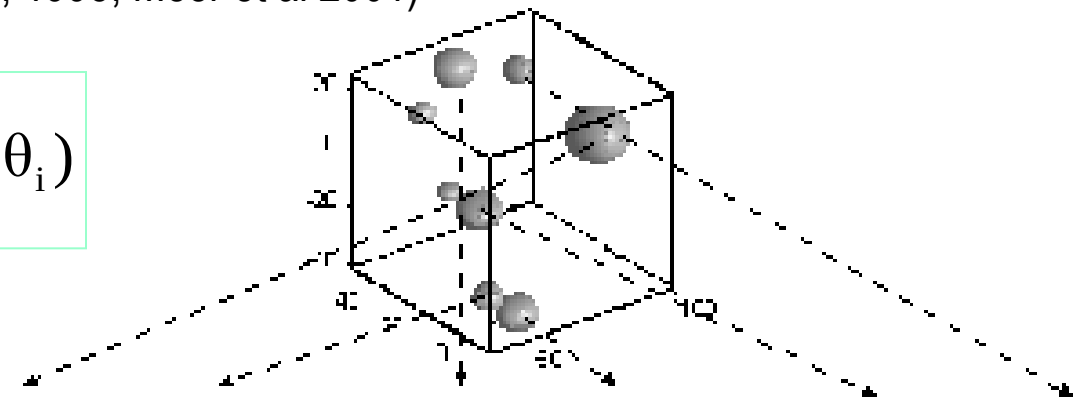

texture



Clustering in Color Space Ω_{c1}

Mean-shift clustering (Cheng, 1995, Meer et al 2001)

$$q(\theta|I) = \sum_{i=1}^{K} \omega_i \, g(\theta - \theta_i)$$



Input

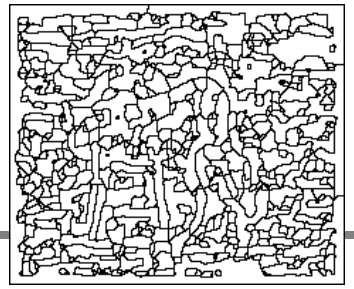
saliency maps 1 2 3 4 5 6

The brightness represents how likely a pixel belongs to a cluster.

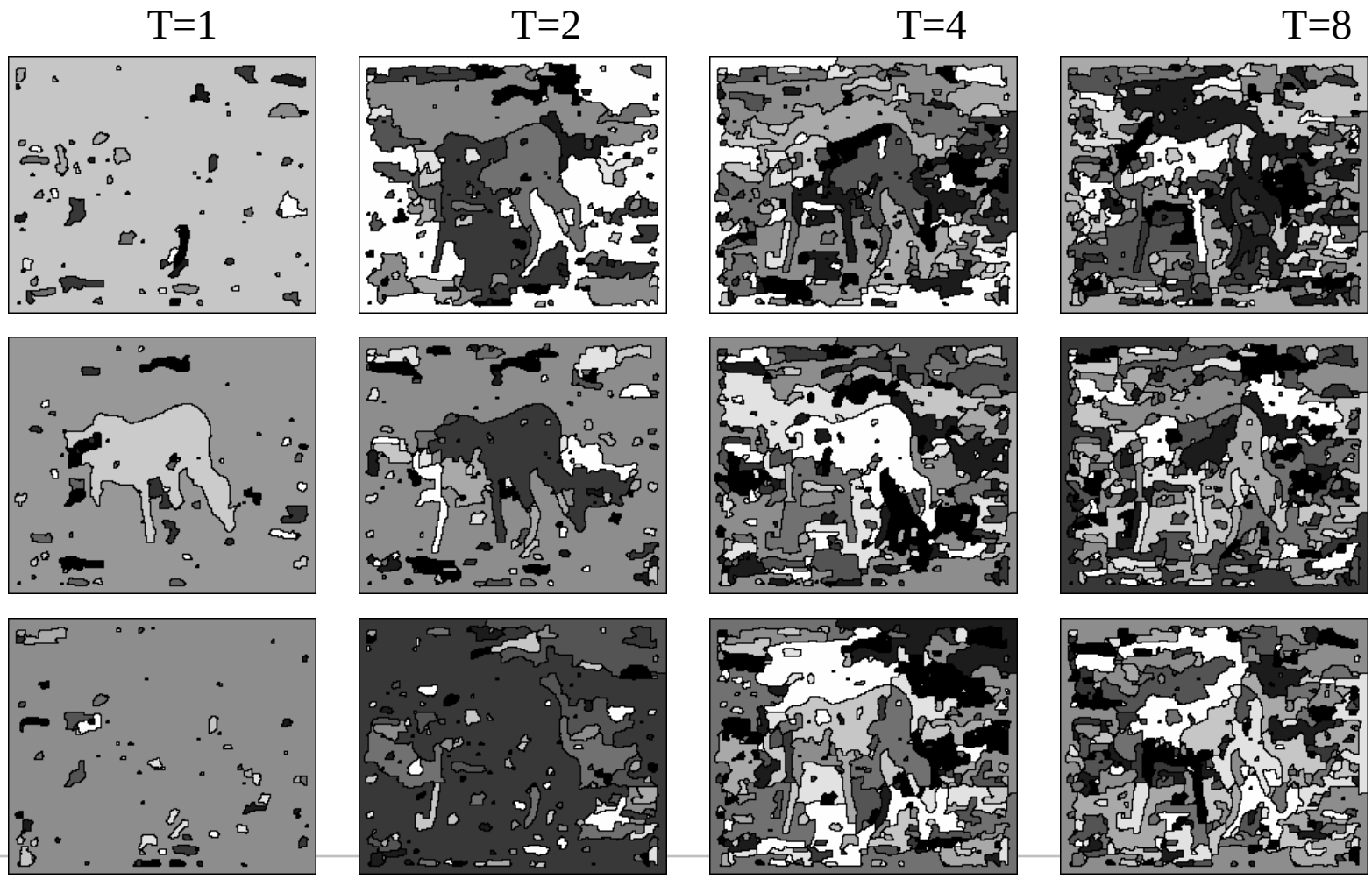
Edges in Partition Space Ω_π

Edge detection and tracing at three scales of details:

a). input b). scale 1 c). scale 2 d). scale 3



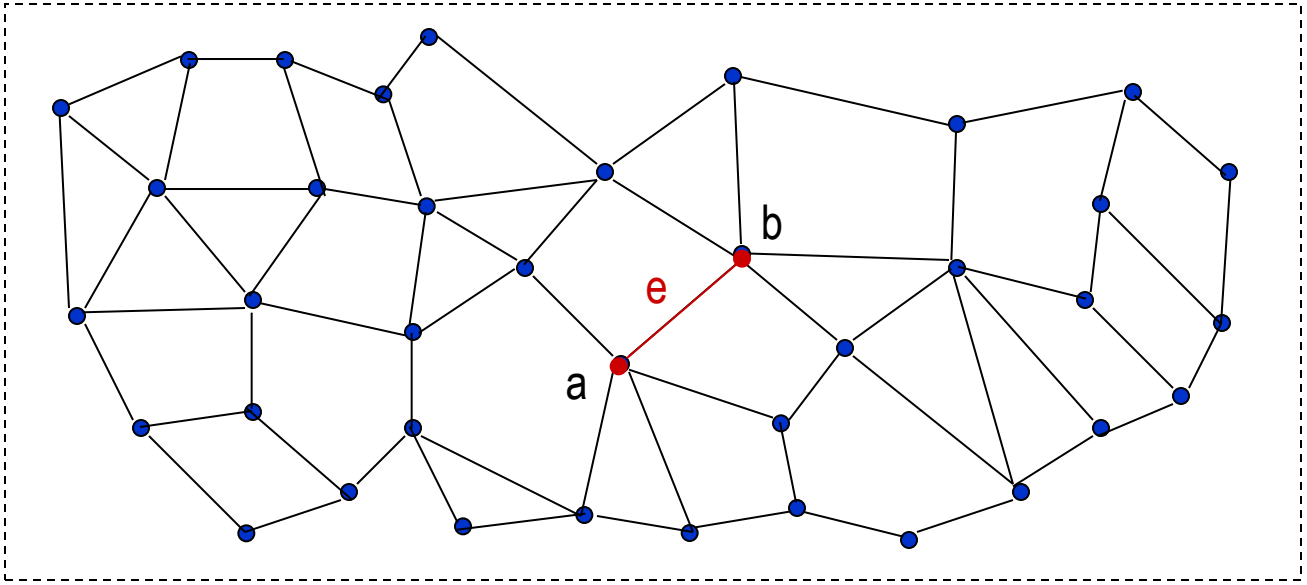
Super-pixels and connected components



Clustering in the Partition Space

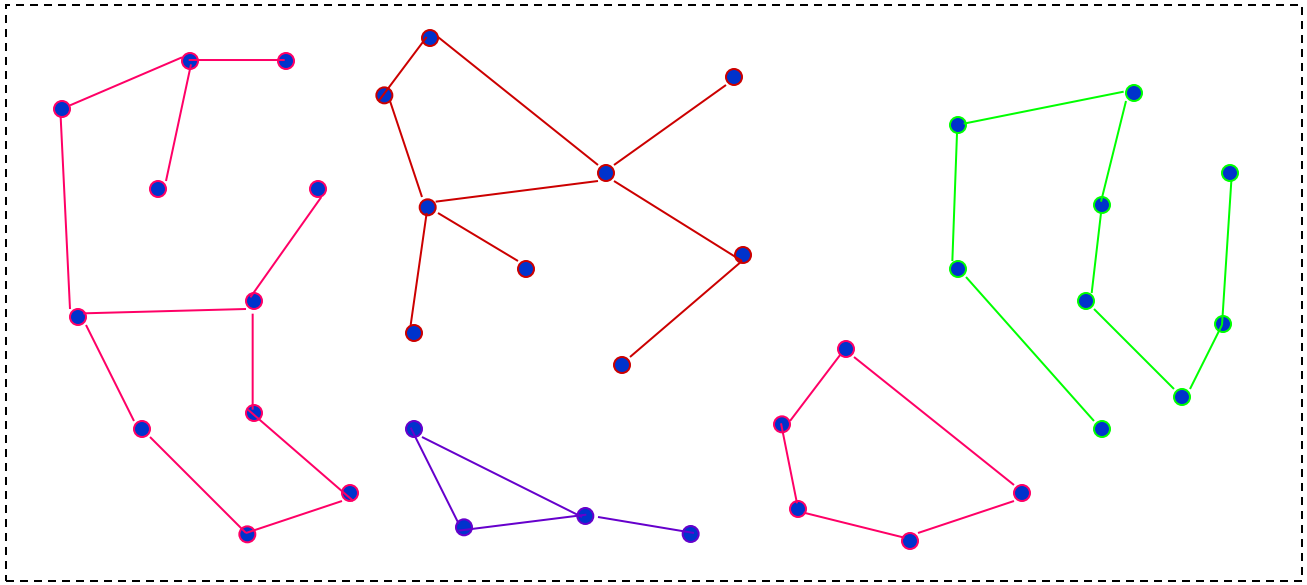
an adjacency graph: each vertex is a basic element : pixels, small-regions, edges, ….
each link e=<a, b> is associated with a probability/ratio for similarity

$$\frac{q(e = \text{"on"} \mid F(I(a)), F(I(b)))}{q(e = \text{"off"} \mid F(I(a)), F(I(b)))}$$



Clustering in the Partition Space

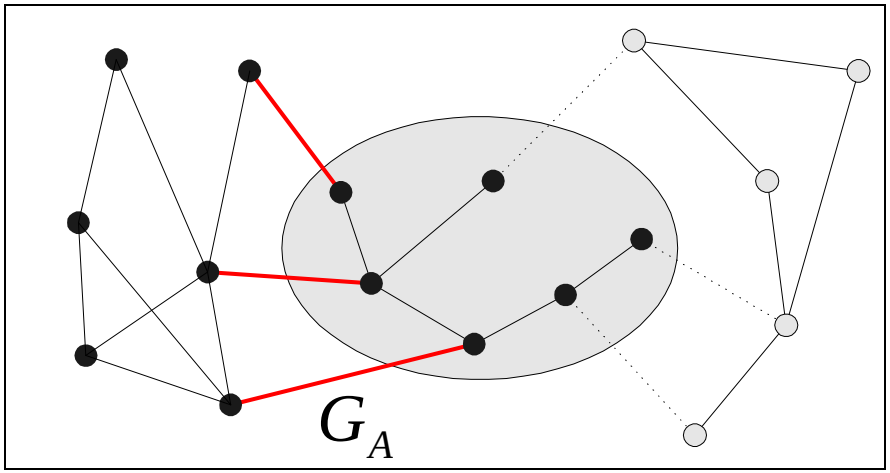
Sampling the edges independently, we get connected components:

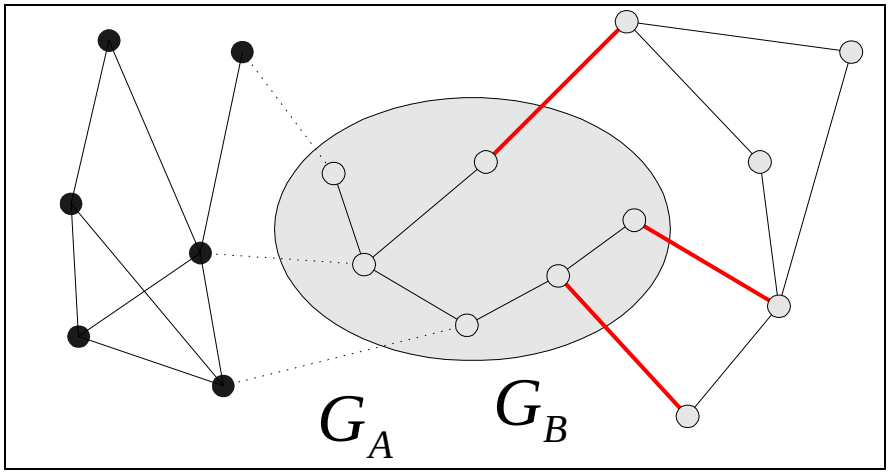


These connected sub-graphs are the clusters in the partition space

sampling $c \sim q(C \mid F(I))$ on Ω_π

Graph Partitioning – Generalizing SW





The red edges are the bridges $E(V^c, V_l - V^c), E(V^c, V_{l'} - V^c)$

Theorem Accepting the label change proposal with probability:

$$\alpha = \min\{1, \frac{p(B)}{p(A)} \frac{p(l|V^c, B, G) \prod_{e \in E(V^c, V_{l'} - V^c)} (1 - q_e)}{p(l'|V^c, A, G) \prod_{e \in E(V^c, V_l - V^c)} (1 - q_e)}\}$$

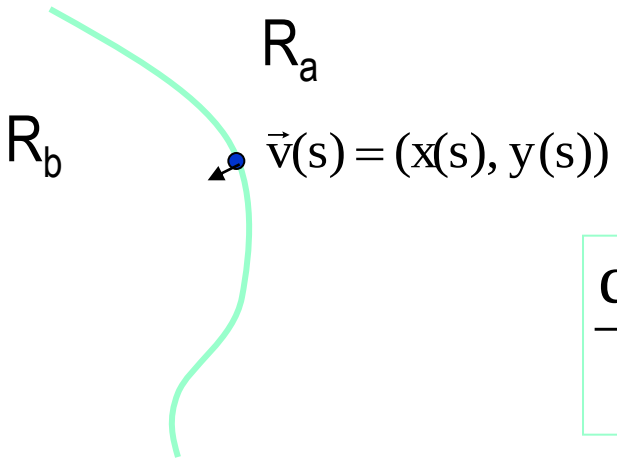
results in an ergodic and reversible Markov Chain.

Diffusion Processes on the boundary

The Markov chains realized reversible jumps between sub-spaces of varying dimensions.

Within a subspace of fixed dimension, there are various diffusion processes expressed as partial differential equations.

For example, the region competition for curve evolution (Zhu, Lee, and Yuille, 95)

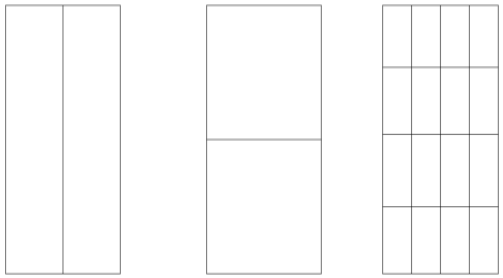


Let v be a point on the boundary between two regions, its motion is governed by the region-competition equation.

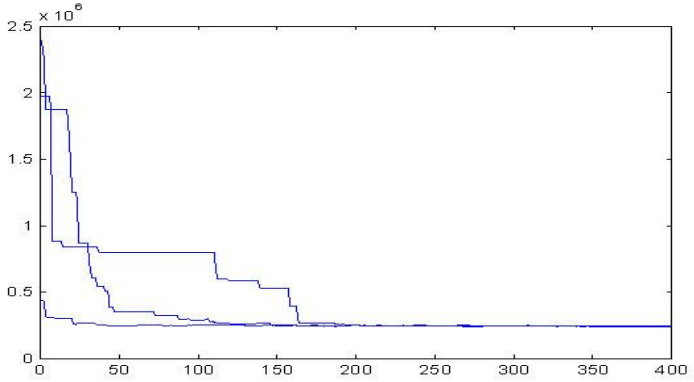
$$\frac{d\vec{v}(s)}{dt} = (\mu \cdot \kappa(s) + \frac{\log p(I(x,y)|\,\theta_a)}{\log p(I(x,y)|\,\theta_b)}) \cdot \vec{n}(s)$$

Running DDMCMC

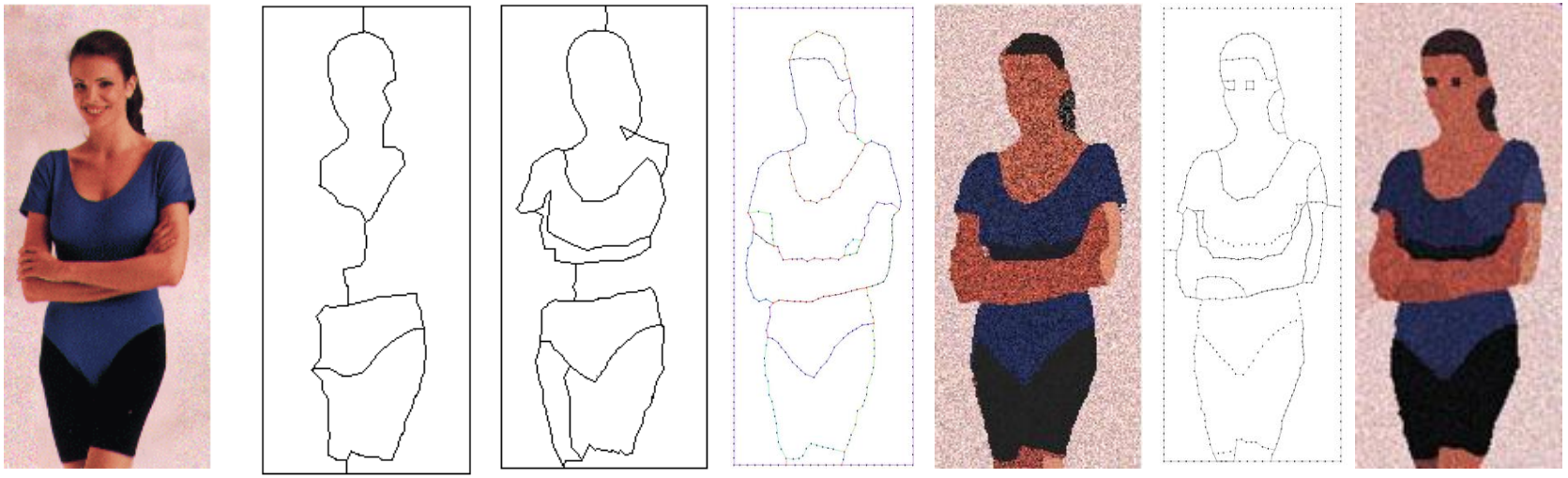
starting with 3 different initial segments below



MC 1 MC 2 MC 3



energy plots of three MCMCs

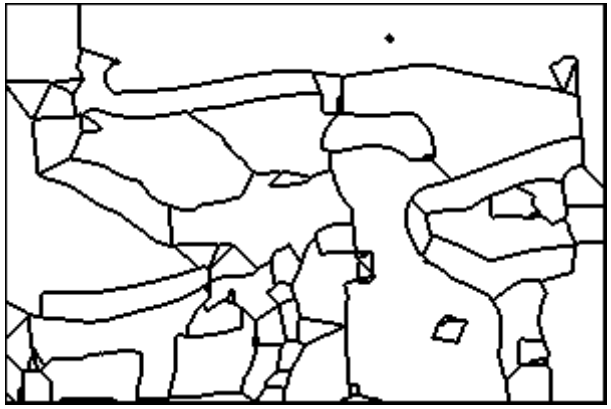


input

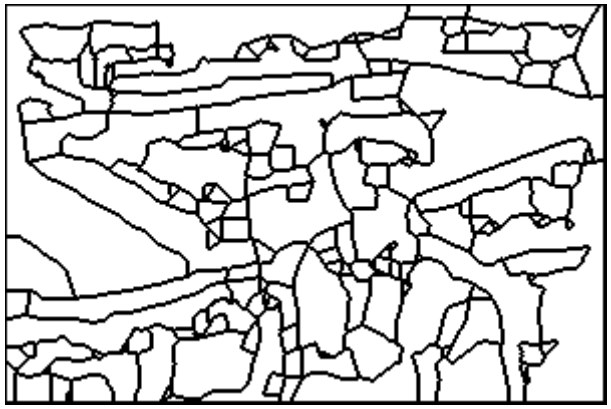
W1 $I_1 \sim p(I|W_1)$ W_2 $I_2 \sim p(I|W_2)$

Proposals by Edge Detection at Different Scales

Partition maps: $q(partition|F_{edge}(\mathbf{I}))$



Scale 1

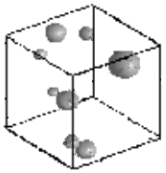


Scale 2

Scale 3

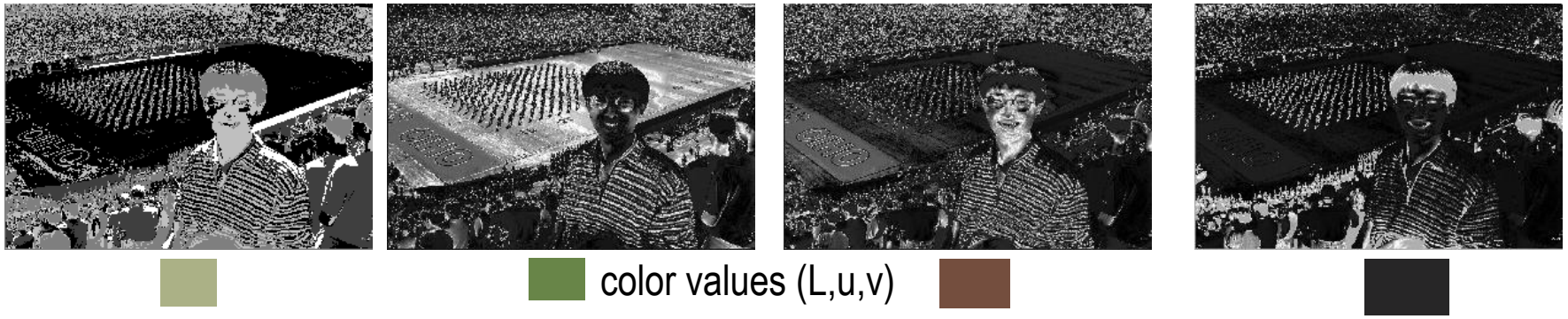
Proposals for region models by clustering

$$q(\theta|F_c(\mathbf{I})) = \sum_{i=1}^{K} \omega_i G(\theta - \theta_i)$$

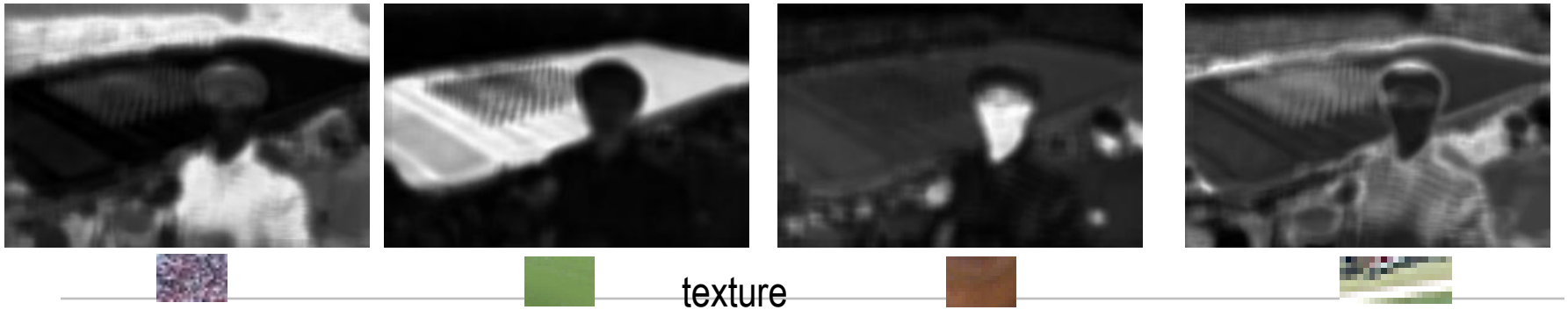


Saliency maps (the brightness represents how likely a pixel belongs to a cluster.)

q_{θ_1} q_{θ_2} q_{θ_3} q_{θ_4}

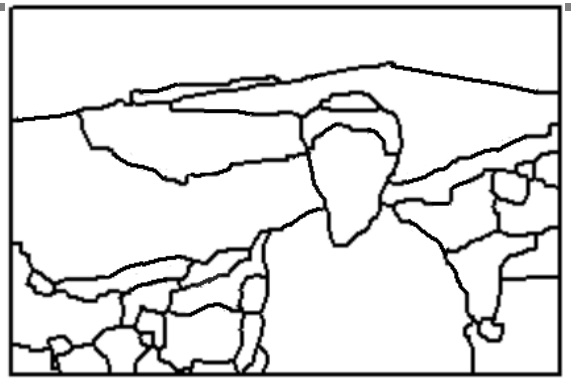


color values (L,u,v)



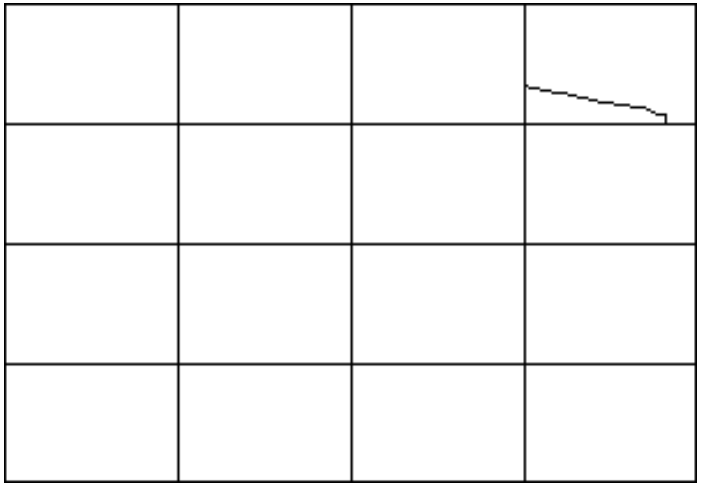
texture

A Demo



Segmentation

Synthesis

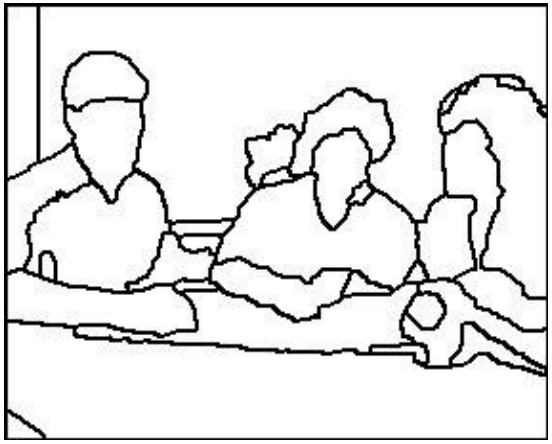


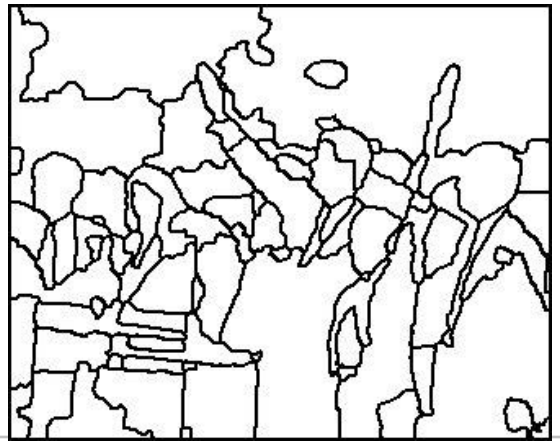
snapshot of solution W

sampled by DDMCMC

Experiments: Color Image Segmentation

Input segment π^* synthesis $I \sim p(I \mid W^*)$

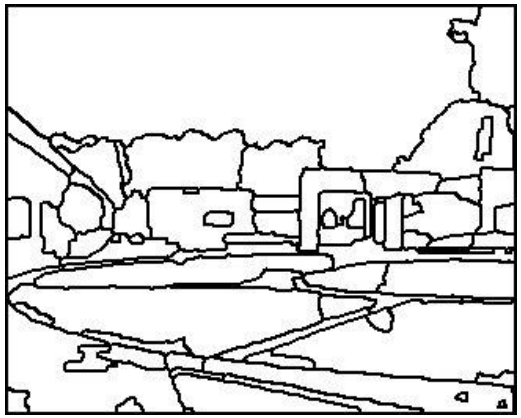
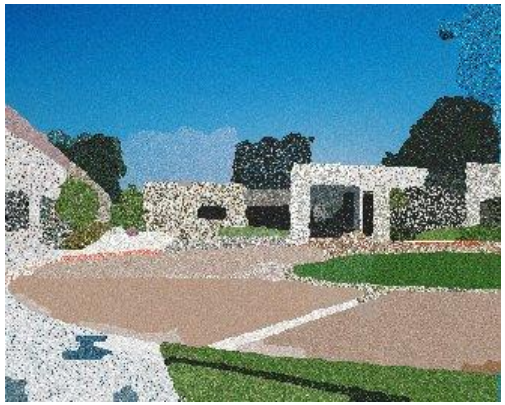



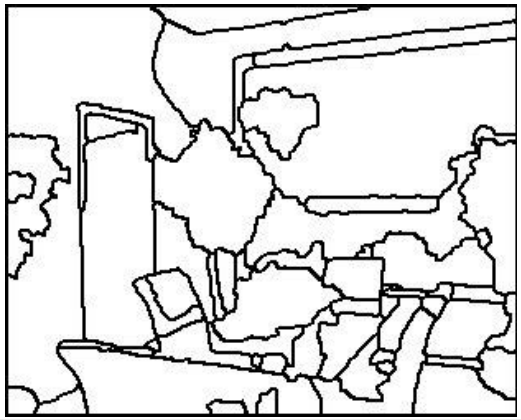

Experiments: Color Image Segmentation

Input segment π^* synthesis $I \sim p(I \mid W^*)$

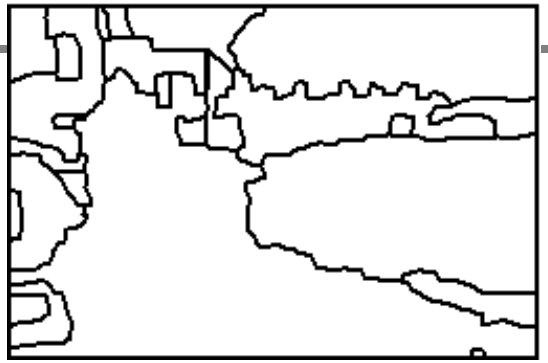
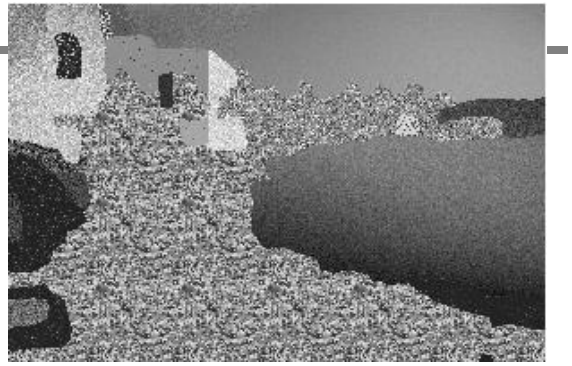
Experiments: Color Image Segmentation

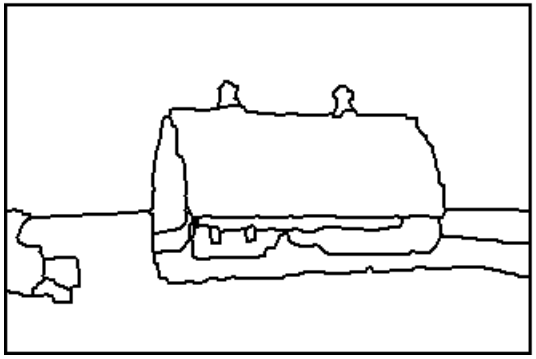
Input | segment π^* | synthesis $I \sim p(I \mid W^*)$




a. Input image b. segmented regions c. synthesis $I \sim p(I \mid W^*)$



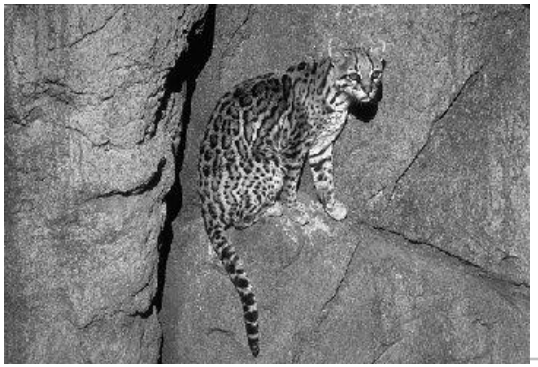
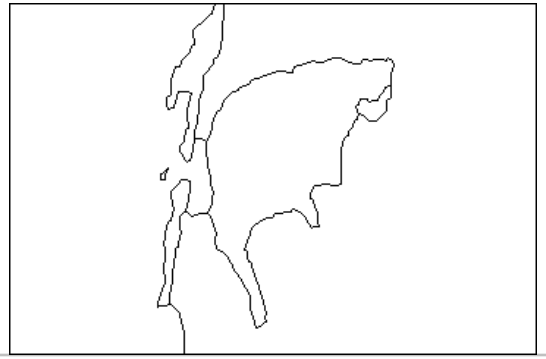
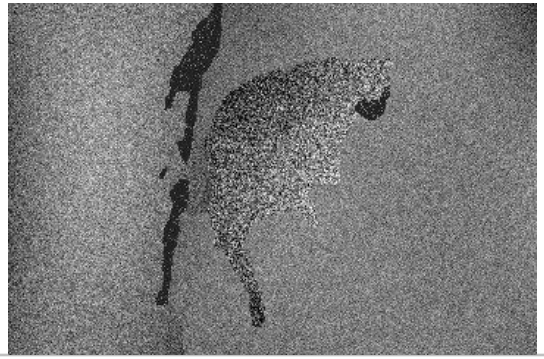
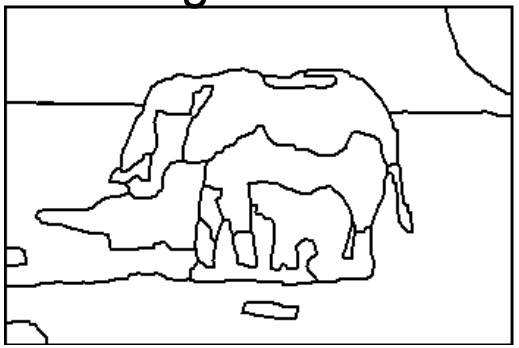




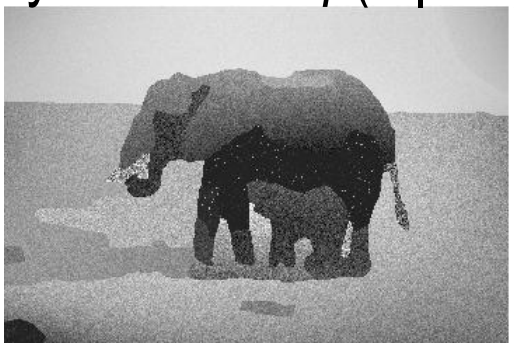
Image Segmentation

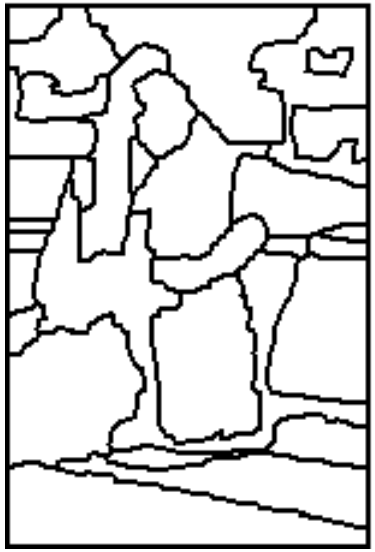
Input

segment π^*



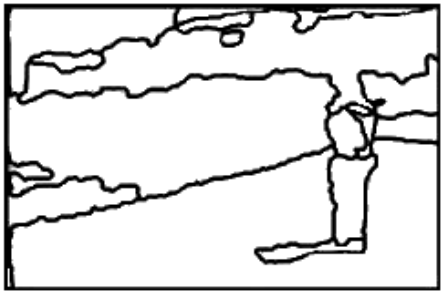
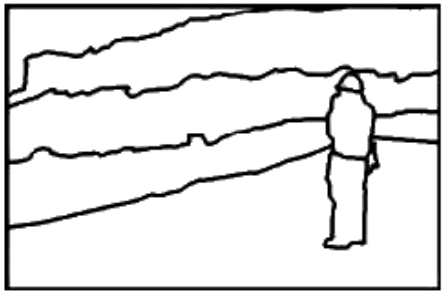
synthesis $I \sim p(I \mid W^*)$





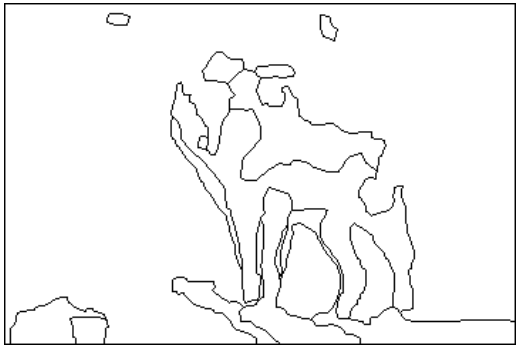
Performance on the Berkeley Benchmark Study

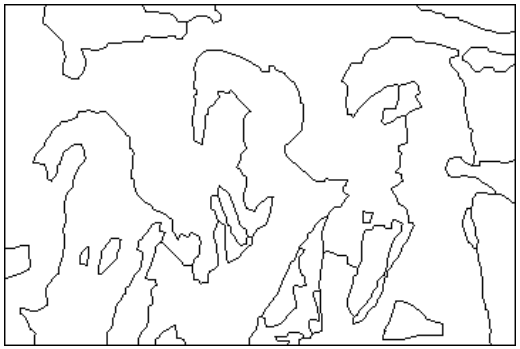
(David Martin et al, 2001)

test images	DDMCMC	manual segment	"error" measure
			0.1083
			0.3082
			0.5627

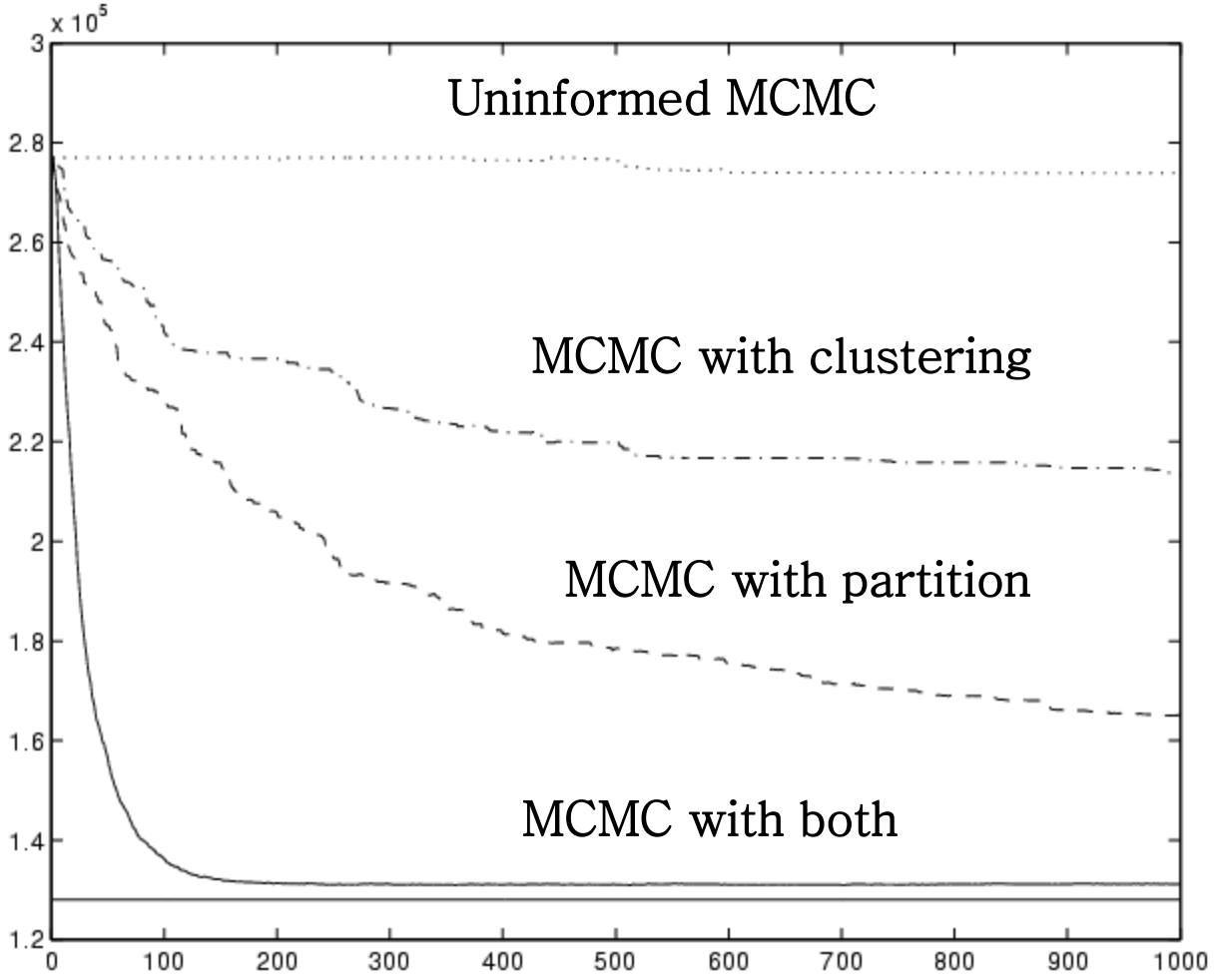
Examples of Failure

a. Input image b. segmented regions c. synthesis $I \sim p(I \mid W^*)$

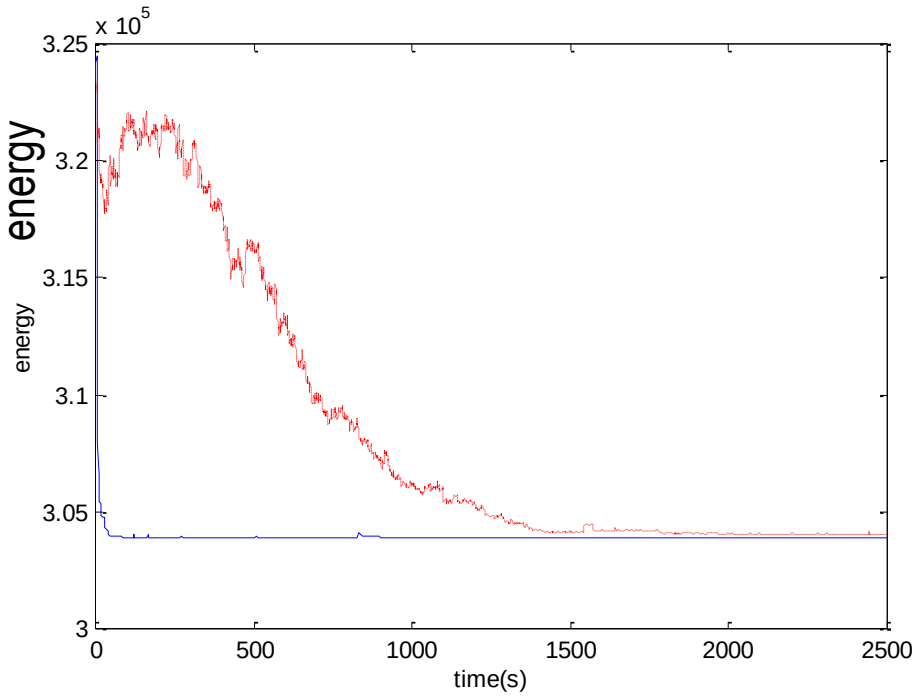




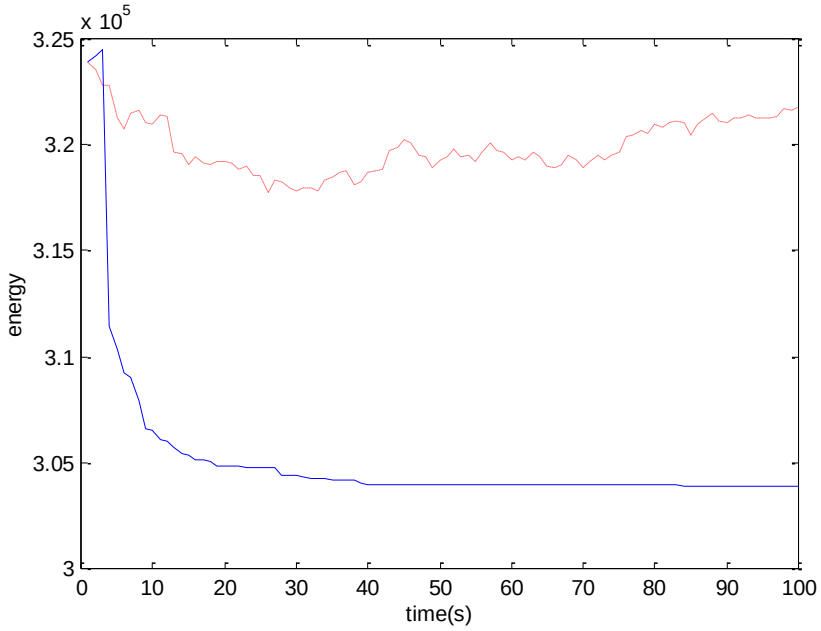
Speed Comparison



Running Time Comparison Against Gibbs Sampler



Time = #sweeps



Zoom-in view

Red curve: Gibbs sampler for graph partition and labeling

Blue curve: Improved SW algorithm for graph partition.

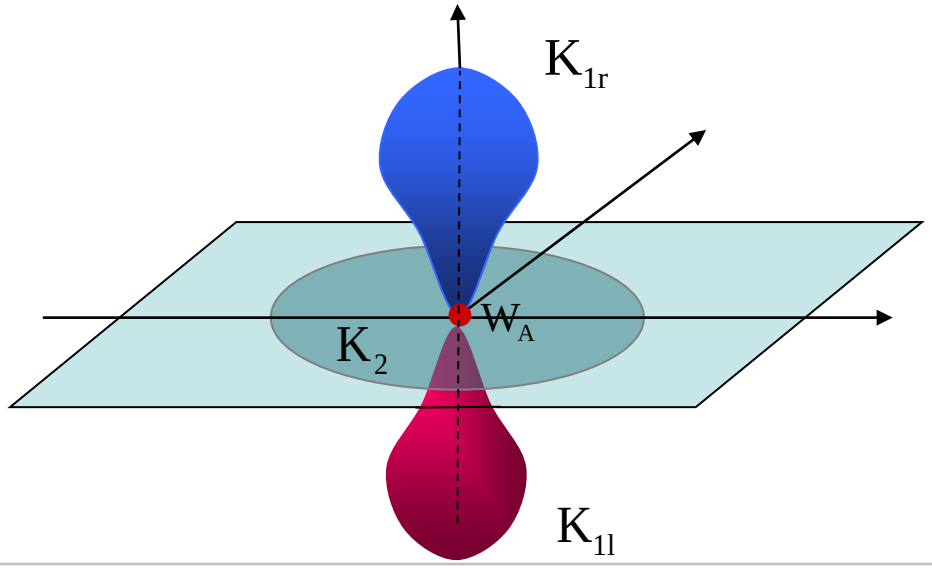
MCMC Kernels consist of many components

Transition kernel is a mixture of many sub-kernels corresponding to the various operators.

$$K(W_A \to W_B) = \sum_a q_a K_a(W_A \to W_B)$$

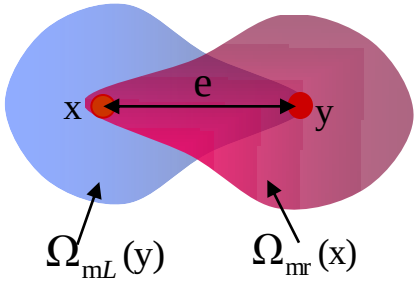
Each observes detailed balance equation, but may not be irreducible.

$$p(W_A)K_a(W_A \to W_B) = p(W_B)K_a(W_B \to W_A)$$



Metropolised Gibbs sampler revisited

Consider a pair of reversible jumps $\mathbf{J}_m$ between x and y.



Proposal according to the conditional probabilities --- like a Gibbs sampler

$$Q_{mr}(x,y) = \frac{\pi(y)}{\sum_{y' \in \Omega_{mr(x)}} \pi(y')}, \qquad y \in \Omega_{mr}(x);$$

$$Q_{ml}(y,x) = \frac{\pi(x)}{\sum_{x' \in \Omega_{ml(y)}} \pi(x')}, \qquad x \in \Omega_{ml}(y);$$

Proposal matrix Q

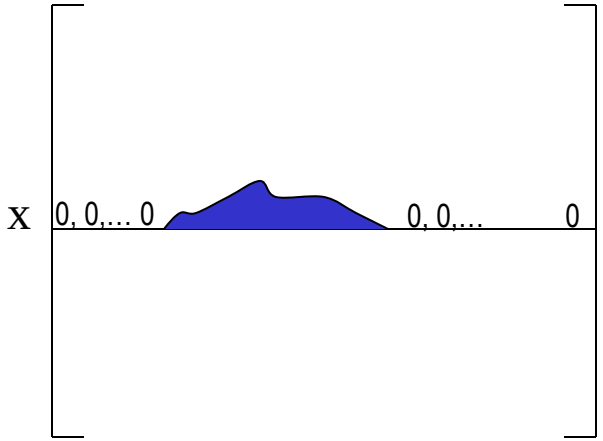
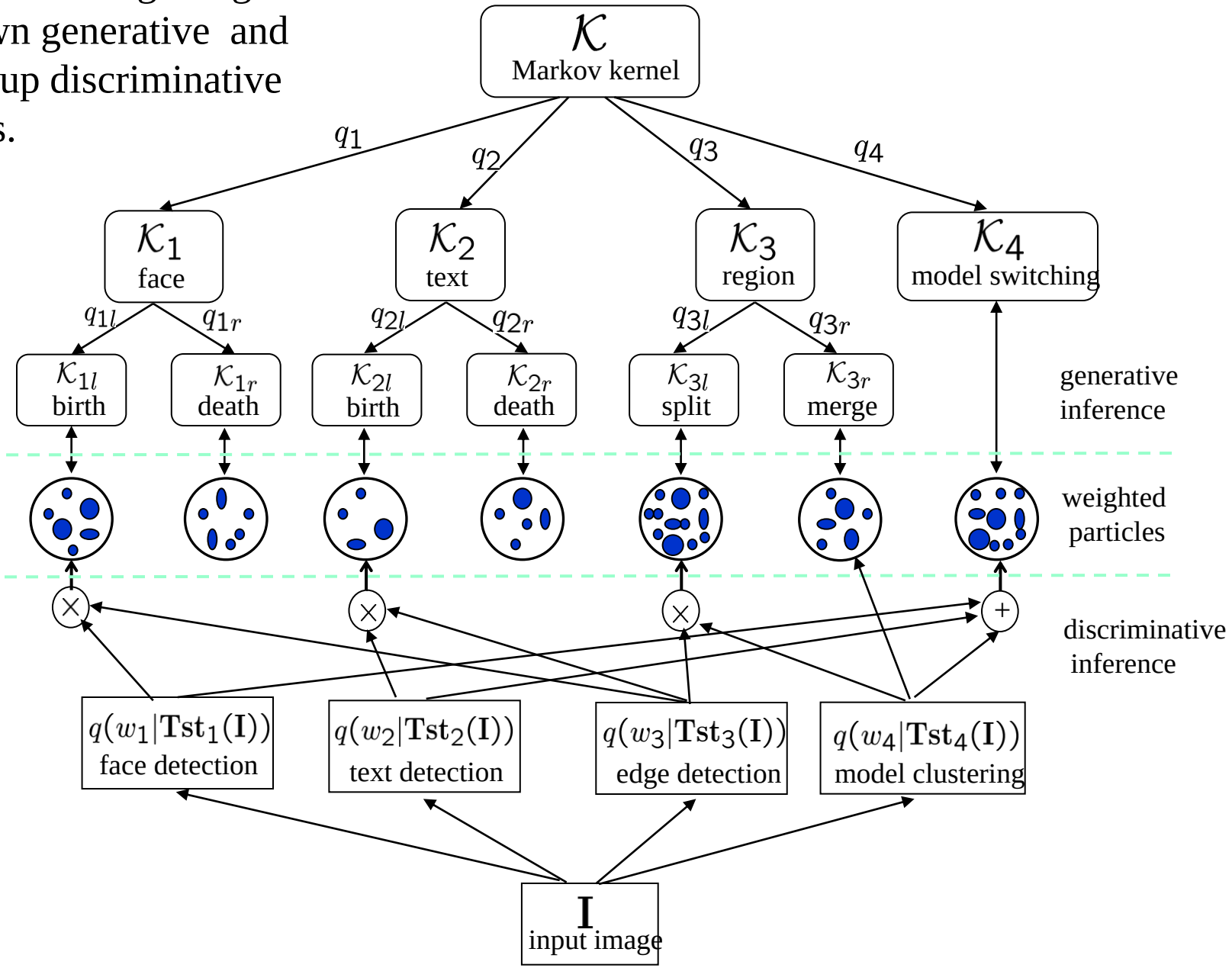


Diagram for Integrating Top-down generative and Bottom-up discriminative Methods.



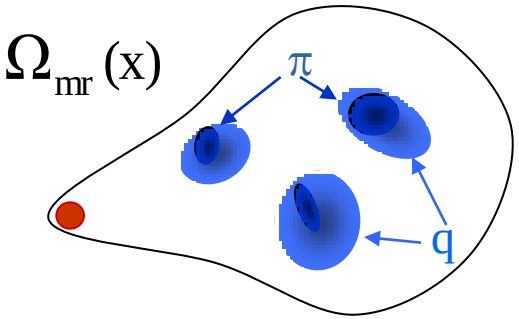
Data-Driven Methods in the object spaces

$$K_{mr}(x,y)=\pi(y)\min\left(\frac{1}{\sum_{y'\in\Omega_{mr}(x)}\pi(y')},\frac{1}{\sum_{x'\in\Omega_{ml}(y)}\pi(x')}\right)$$

We conjecture that the Metropolised-Gibbs sampler is the best design strategy on average ---- mixing very fast under the constraints of scopes.

But at each step, it need to evaluate the expensive posterior probability over a rather large scope

$$\Omega_{mr}(x)\cup\Omega_{ml}(y)$$



We replace the condition probability by bottom-up (discriminative) methods which are estimated locally with lower cost. We show that such approximations indeed reduce mixing time.

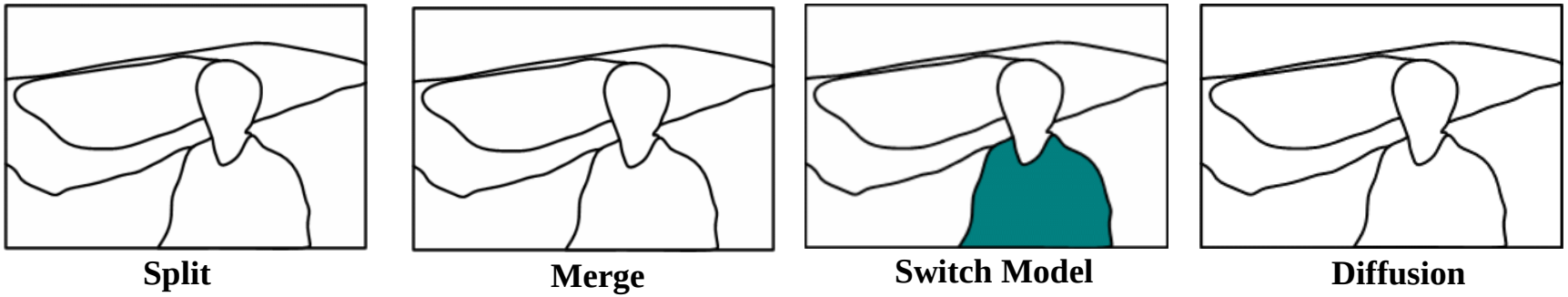
MCMC Moves (Jumps and Diffusions)

K_{1l}: Splitting of a region into two.

K_{1r}: Merging two regions into one.

K_2: Switching the model type for a region.

K_3: Diffusion of region boundary -- region competition (Zhu and Yuille 1996).



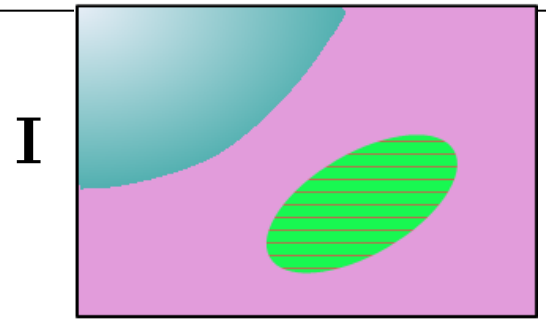
It integrates a variety of existing segmentation methods in computer vision such as:
Edge Detection, Clustering, Split-merge, Region Competition (Snake, MDL, PDE)…

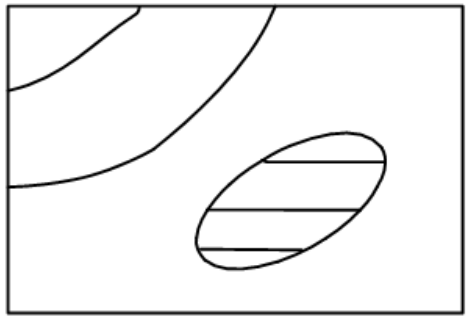
Split and Merge

Consider a reversible jump: $W_A = (1, (R_1, l_1, \theta_1)) \leftrightarrow W_B = (2, (R_2, l_2, \theta_2), (R_3, l_3, \theta_3))$

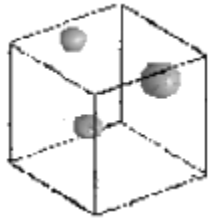
$$\underbrace{K(W_A \rightarrow W_B)}_{transition\ probability} = \underbrace{Q(W_B|W_A; F(\mathbf{I}))}_{proposal} \underbrace{\alpha(W_A \rightarrow W_B)}_{acceptance\ rate}$$

$$Q(W_B|W_A, F(\mathbf{I})) = Q(R_2, R_3|W_A; F_{edge}(\mathbf{I})) \cdot Q(l_2, \theta_2|R_2, W_A; F_c(\mathbf{I})) \cdot Q(l_3, \theta_3|R_3, W_A; F_c(\mathbf{I}))$$

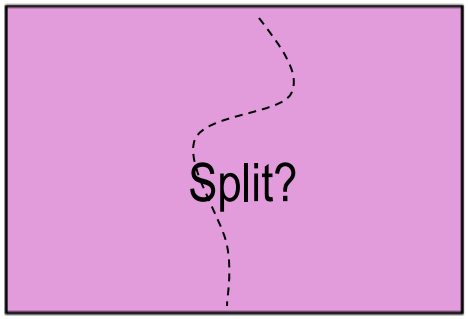




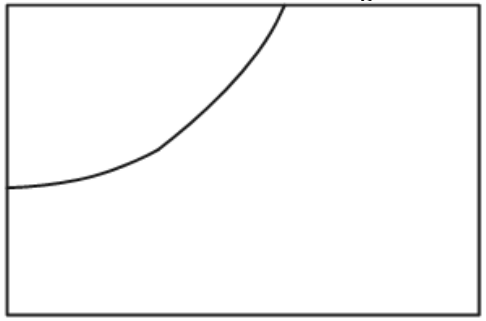
$q(partition|F_{edge}(I))$



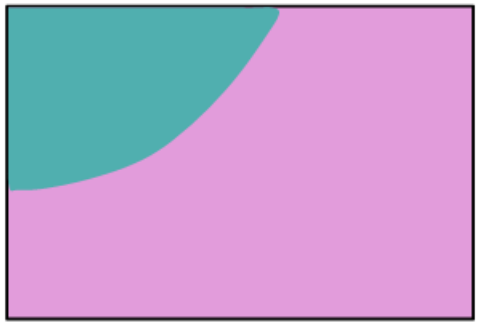
$q(\theta|F_c(I))$



W_A

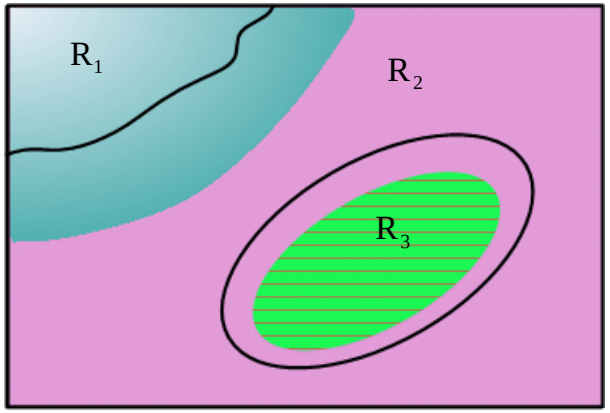


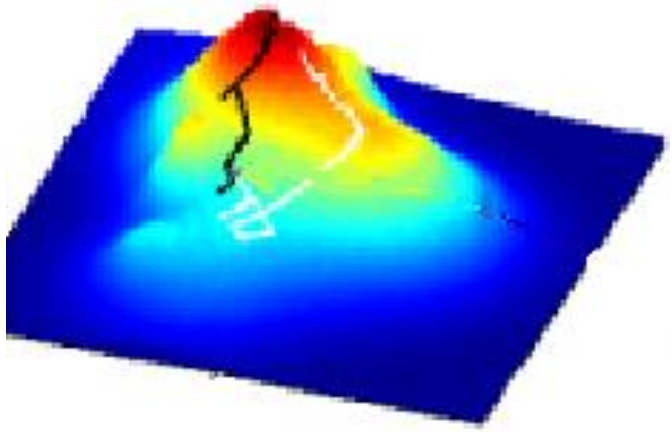
$Q(R_2, R_3|W_A; F_{edge}(I))$



W_B

Stochastic Diffusion and PDE





The continuous Langevin equation simulates a Markov Chain with stationary density

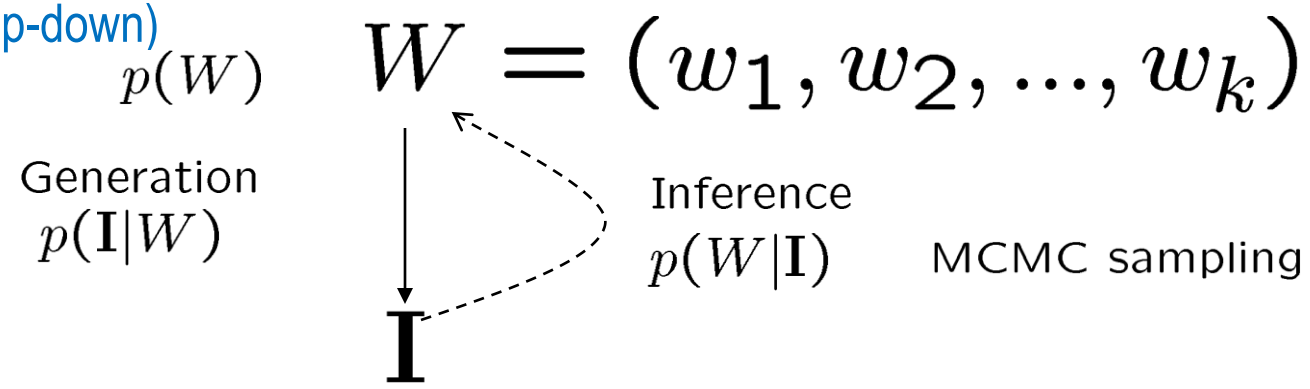
$$p(W|\mathbf{I}) \propto exp\{-E(W)/T\}$$

For example, the movement of changing point is driven by

$$\frac{dx(t)}{dt} = \{[\log p(\mathbf{I}(x)|l_i,\theta_i) - \log p(\mathbf{I}(x)|l_j,\theta_j)] - \kappa_x + \sqrt{2T(t)}N(0,1)\}\vec{n}$$

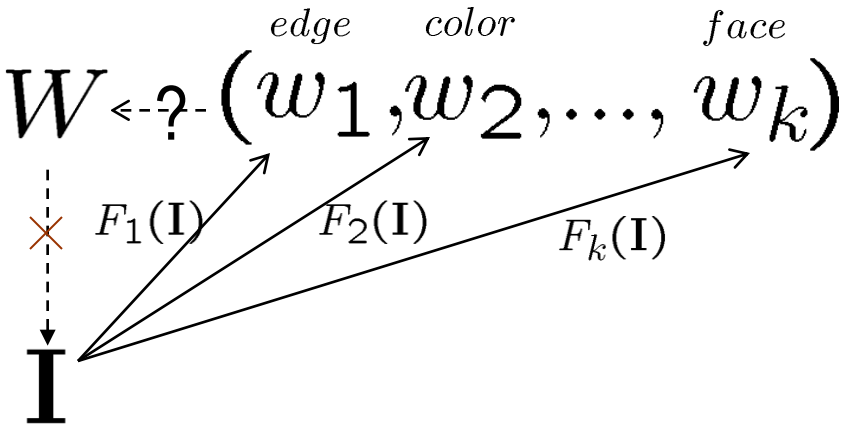
Summary: generative vs. discriminative

Bayesian: (Top-down)



$$W^* = \arg\max\ p(W|\mathbf{I}) = \arg\max\ p(\mathbf{I}|W)p(W)$$

Data-driven: (Bottom-up)



$$q(w_j|F_j(\mathbf{I})) \rightarrow p(w_j|\mathbf{I}), j = 1...k$$

From segmentation to parsing

$$W = (n, \{(\zeta_i, R_i, l_i, \theta_i), i = 1, ...n\})$$

Face images of FERET dataset

Text images of San Francisco street scenes.

Generic Images parsing

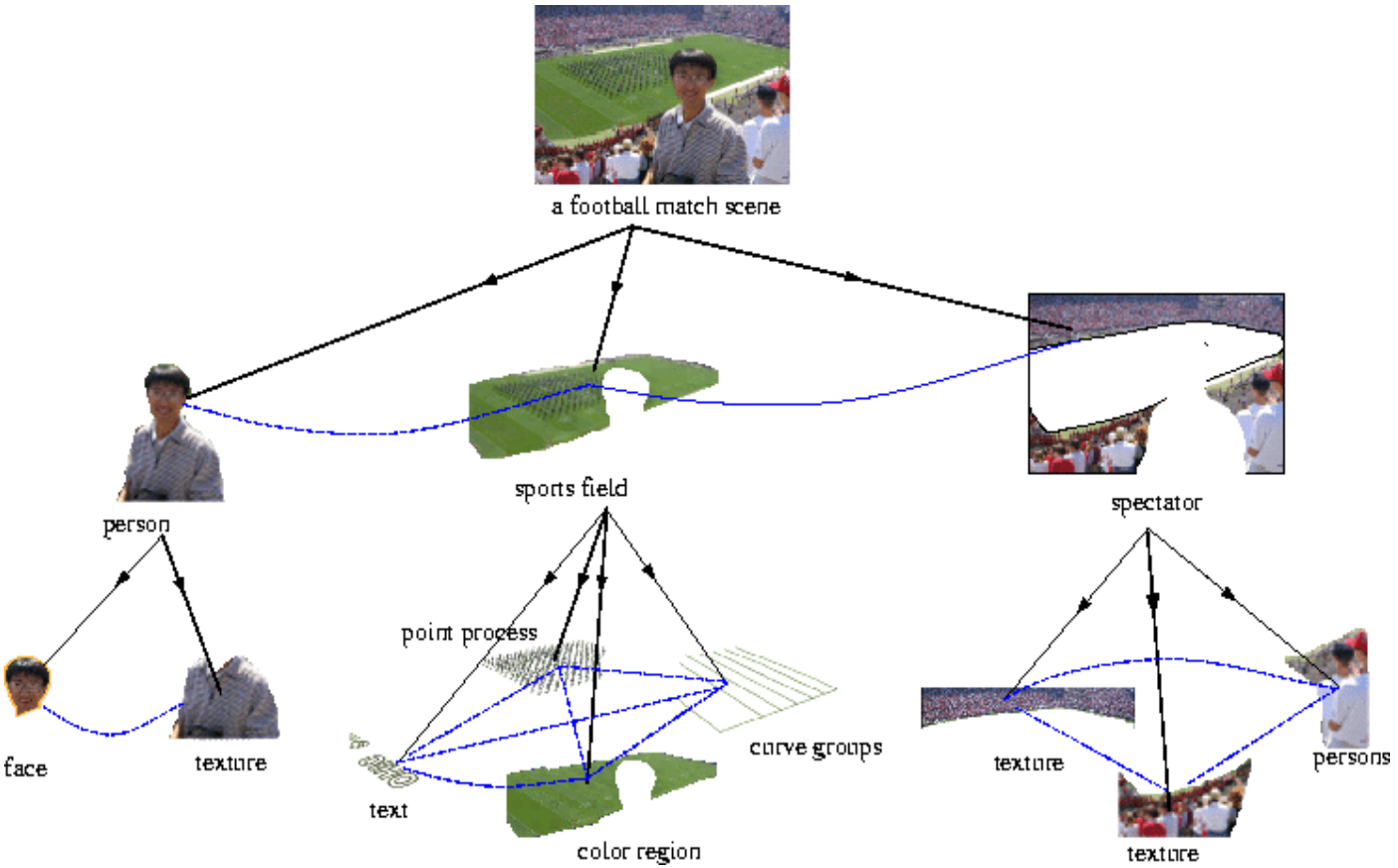
scene

objects

patterns

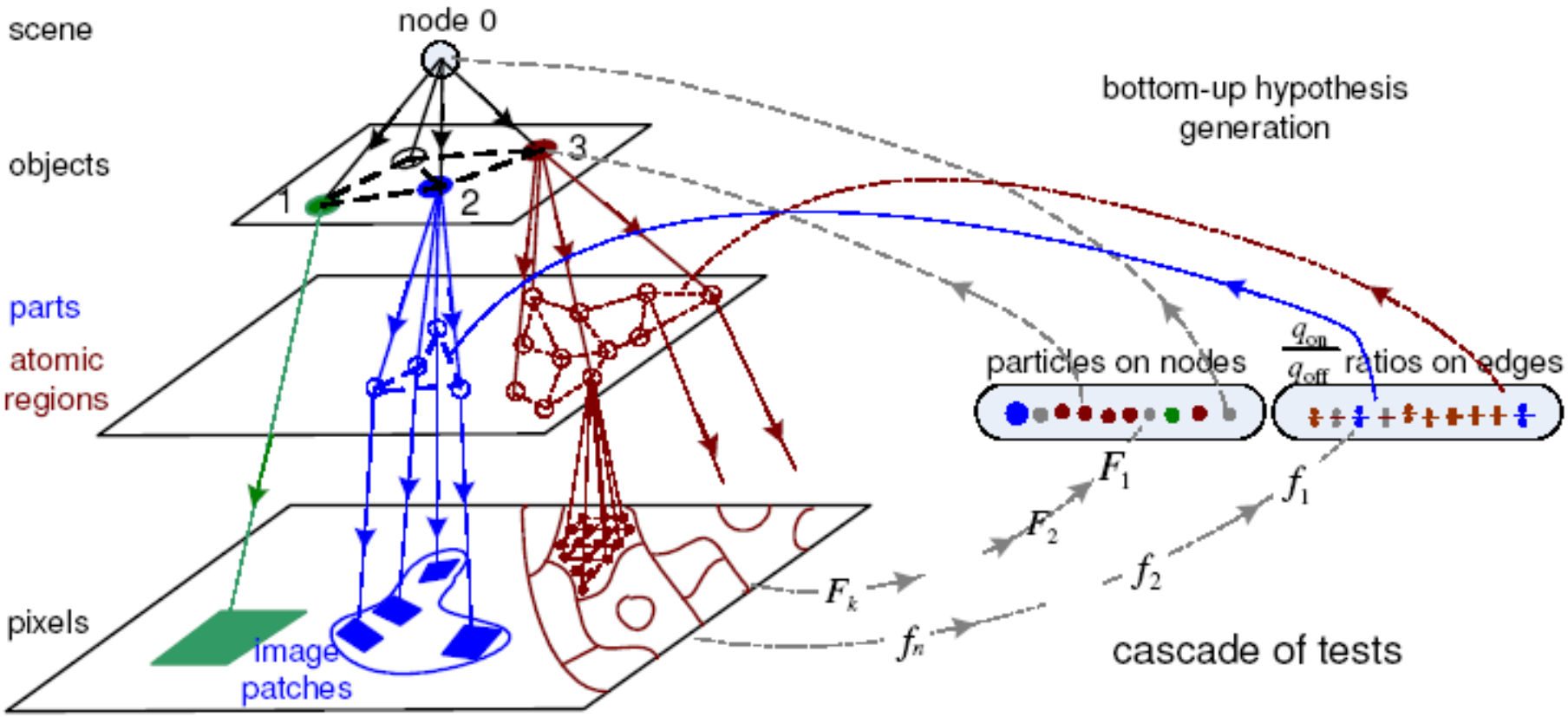
parts

textons



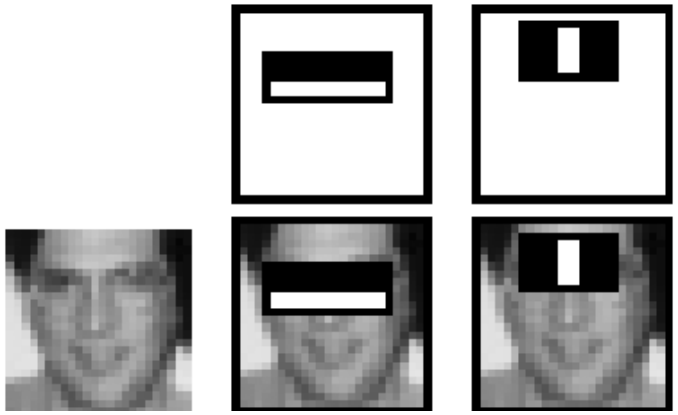
Example: parsing (Tu et al, 2000-2004)

Integrating generative and discriminative

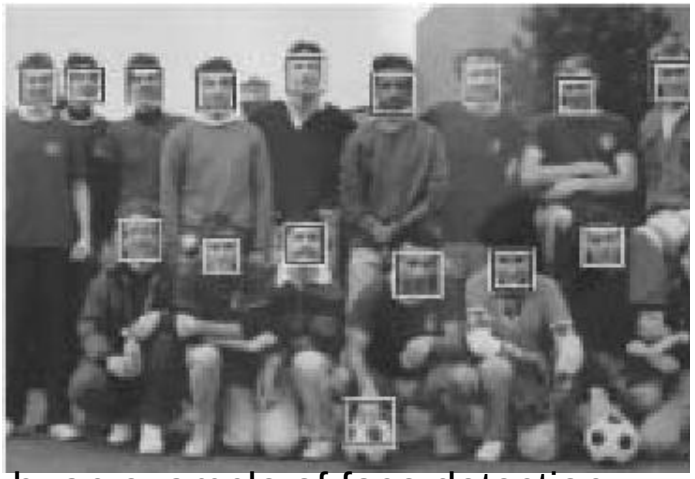


Adaboost in the Label Space

---- an example from Viola and Jones, 2001.



a. the first two face features



b. an example of face detection

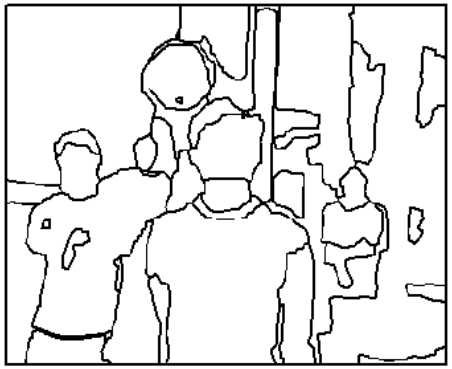
Adaboost is a learning algorithm which makes decision by combining a number of simple features. As T and training samplers become large enough, it weakly converges to the log ratio of the posterior probability.

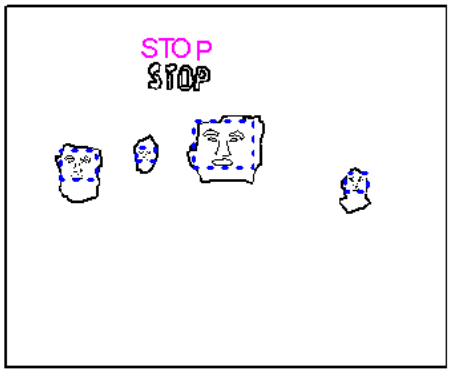
$$y = sign\big(\lambda_1 h_1(x) + \cdots + \lambda_T h_T(x)\big) = sign(\log \frac{p(y=+1|x)}{p(y=-1|x)})$$

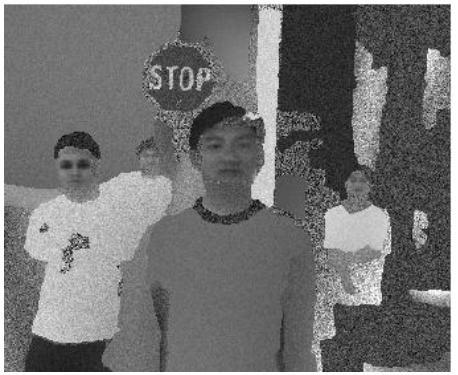
Image Parsing Results

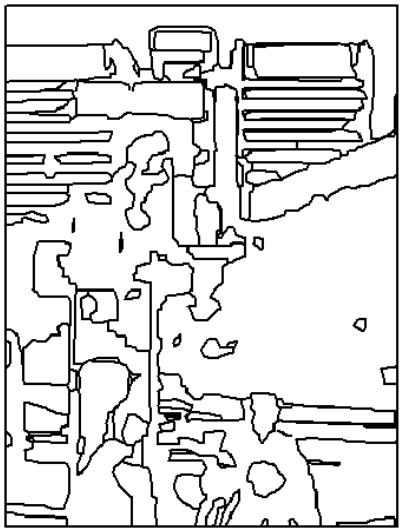
Tu, Chen, Yuille, and Zhu, iccv2003

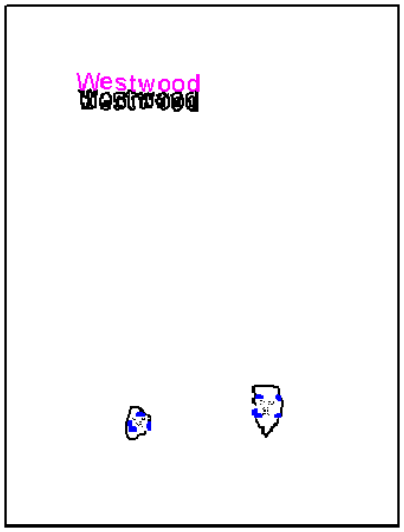
Input | Regions | Objects | Synthesis











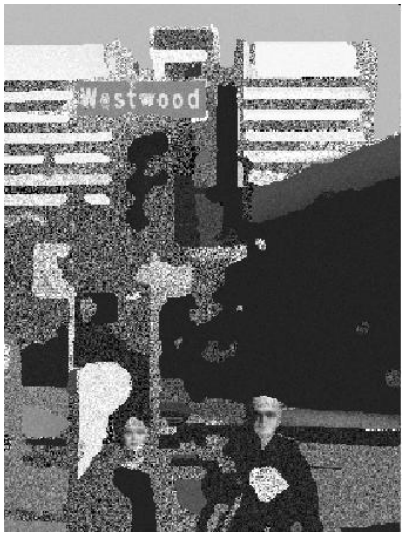
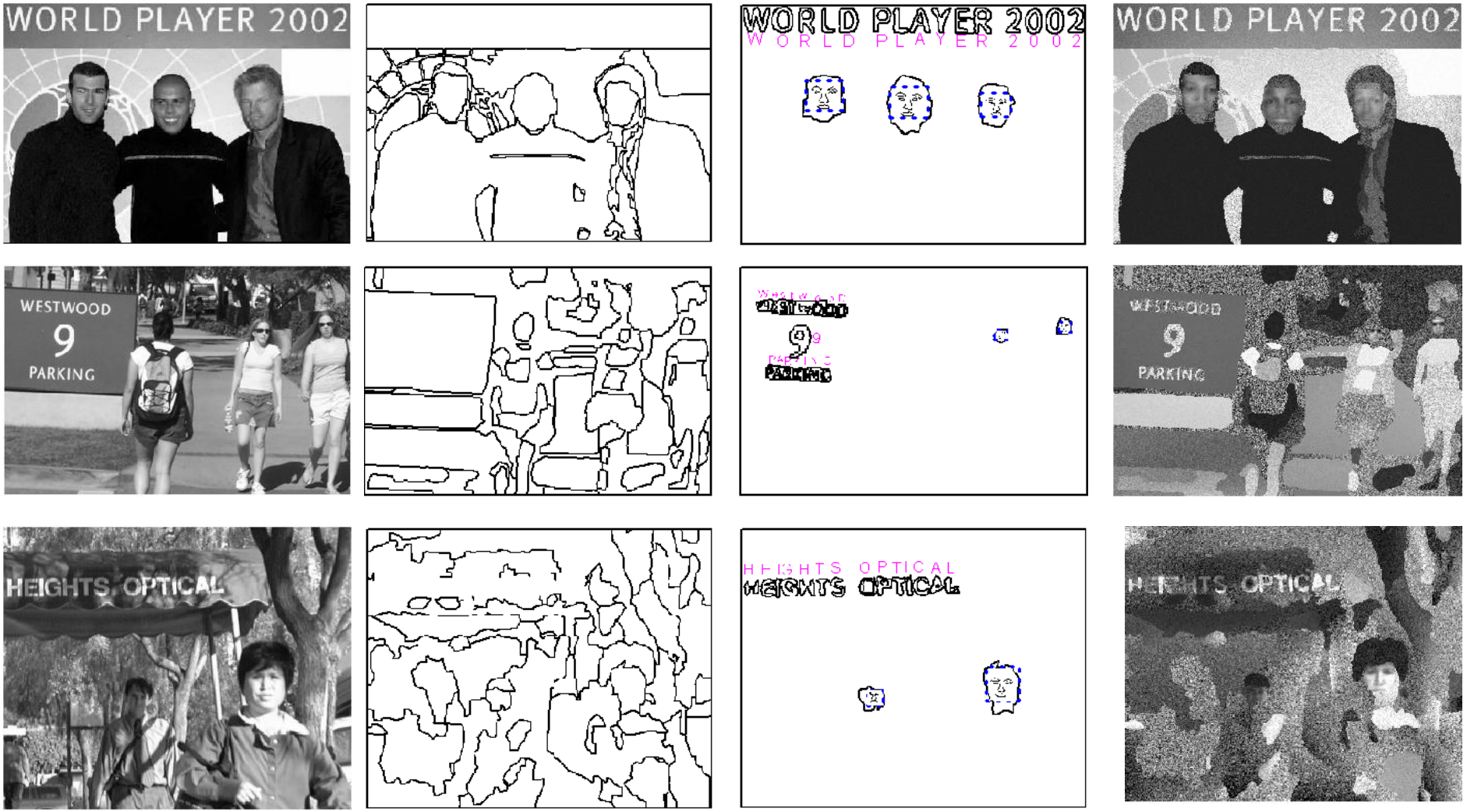


Image Parsing Results

Input | Regions | Objects | Synthesis



Review: Some MCMC developments related to vision