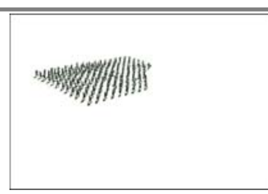
Visual Inference by Markov Chain Monte Carlo Methods

Song-Chun Zhu

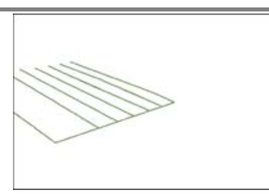
Joint work with Z. Tu, A. Barbu, F. Han, R. Maciuca, A. Chen, A.Yuille

Parsing an Image Into Its constituent Patterns

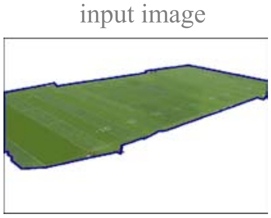
input image



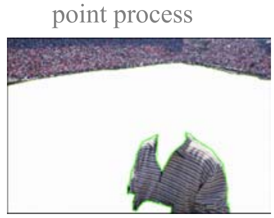
point process



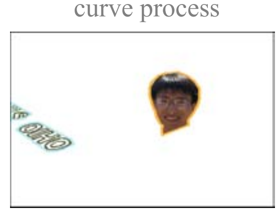
curve process



a color region



texture regions



objects

1. The task integrates conventional vision tasks:
   image segmentation, perceptual organization, object recognition, etc.
2. Many families of generative models compete to explain the image.

Questions

1. How do we coordinate many types of object models robustly ?

2. What is a good computing strategy for integrating “top-down” with “bottom-up”?

3. How do we compute globally optimal solutions, multiple solutions?

We need a theoretical foundation for answering these questions.

A Bayesian Formulation of Vision

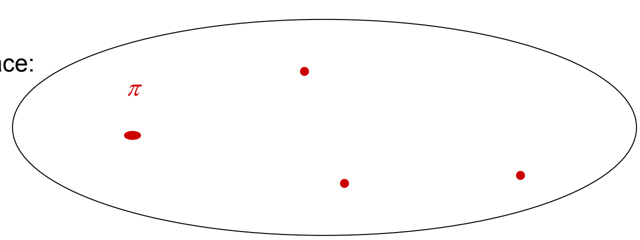
Let I be an image and X be a semantic representation of the world.

$$\boxed{X^* = \arg\max_{X \in \Omega} \pi(X \mid I) = \arg\max_{X \in \Omega} \pi(I \mid X)\pi(X)}$$

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

$$\boxed{(X_1, X_2, \ldots, X_k) \sim \pi(X \mid I)}$$

Search space:

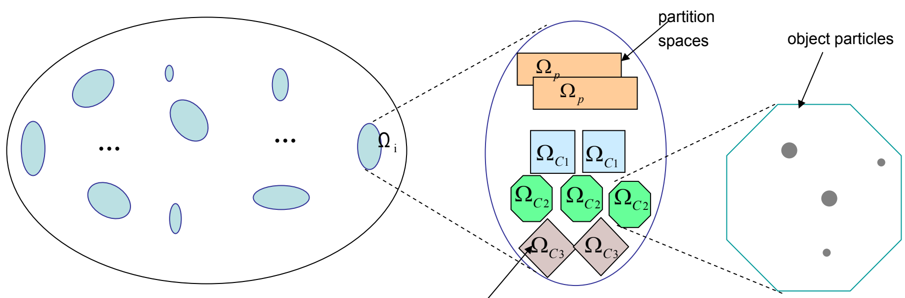


The State Space

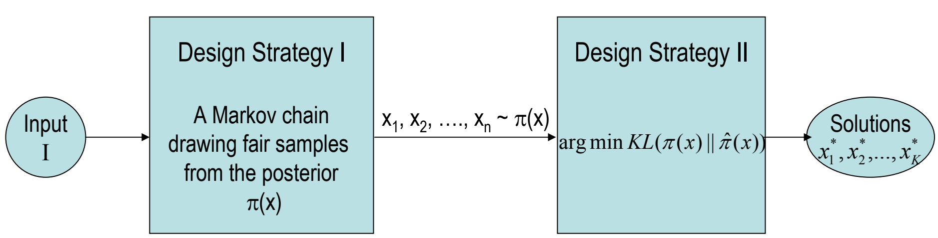
1. Ω has a large number of sub-spaces of varying dimensions.

2. Each sub-space is a product of
   some *partition (coloring) spaces* ---- what go with what?
   x some *object spaces* ---- what are what?

3. The posterior has low entropy, thus the *effective volume* of the search space is relatively small !



Formulation of the Computational Problems



Searching the state space by Markov chain

Markov chain is a triplet MC=(Ω, ν, K)

---- Ω is state space, each state is a solution,
---- $\nu(x_o)$ is probability of initial state,
---- k(x,y) =k(y|x) is the transition probability.

Suppose a MC starts with x_o, after n steps, its state follows a probability,

$$x_n \sim K^n(x_o, x)$$

We wish it gets close to the target $\pi(x)$ as soon as possible.

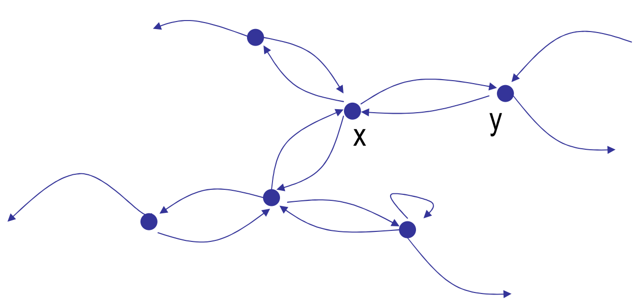
A Theorem

$$d_{TV}(t) = \frac{1}{2}\sum_{x\in\Omega} \| K^n(x_0, x) - \pi(x) \| \le \sqrt{\frac{1-\pi(x_0)}{4\pi(x_0)}} \quad \lambda^n_{slem}$$

$0 < \lambda_{slem} < 1$ Is the second largest eigen value modulus of the transition matrix K.

Searching the state space by Markov chain

The MC kernel K corresponds to a transition graph G for discrete space Ω.



Two concepts:

1. The scope of a state x is a set $\Omega(x) = \{y: \ K(x,y) > 0\}$

2. The capacity of an edge e=(x,y) is $\varphi(e) = \pi(x)K(x,y) = \pi(y)K(y,x)$

Markov Jumps and their scopes

Each state x is connected to some other states by μ pairs of jumps

$\mathbf{J}_m = (\mathbf{J}_{mr}, \mathbf{J}_{ml})$, m=1,2,..., μ

For example, death-birth, split-merge, model switching, grouping-ungrouping etc.

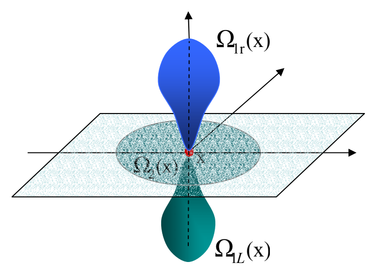
These jumps have their scopes at state x,

$$\Omega_{mr}(x) = \{ y : K_{mr}(x,y) > 0, y \neq x \}$$

$$\Omega_{ml}(x) = \{ y : K_{ml}(x,y) > 0, y \neq x \}$$

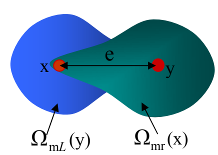
So x is connected to a set,

$$\Omega(x) = \cup_{m=1}^{\mu} (\Omega_{mr} \cup \Omega_{ml})$$



Metropolized Gibbs sampler

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

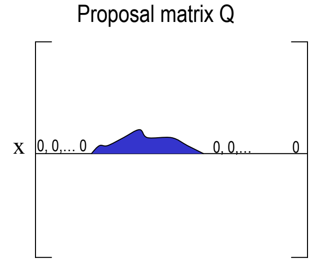


Proposal according to the conditional probabilities --- like a Gibbs sampler

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

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

Proposal matrix Q



Metropolized Gibbs sampler

$$K_{mr}(x, y) = Q_{mr}(y \mid x) \min\left(1, \frac{Q_{ml}(x \mid y)}{Q_{mr}(y \mid x)} \cdot \frac{\pi(y)}{\pi(x)}\right)$$

The Metropolis step corrects the proposal probability ratio by the target probability ratio.

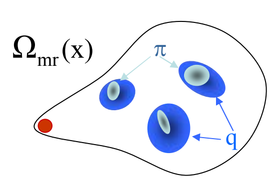
The target prob. ratio follows *generative models*
The proposal prob. ratio follows a factorized form by *discriminative models*

$$\frac{Q_{ml}(x \mid y;\ \ F(I))}{Q_{mr}(y \mid x;\ \ F(I))} \cong \frac{\pi(y \mid I)}{\pi(x \mid I)}$$

Local image features in various model spaces

Data-Driven Methods in the object spaces

Within each jump scope, we replace the condition probability by discriminative models which are estimated locally with lower cost.

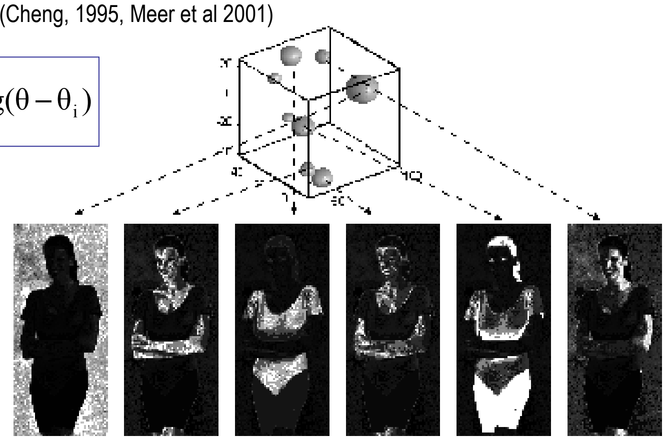


Example 1: Clustering in Color Space

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

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

Input



saliency maps 1 2 3 4 5 6

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

Example 2: Object detection (label) by Boosting method

As the number of features and training examples become large enough, adaboost weakly converges to the log ratio of the posterior probability (Schapire et al).

$$\ell = \mathrm{sign}(a_1 h_1(I) + \cdots + a_1 h_1(I)) \rightarrow \mathrm{sign}\left(\frac{\pi(\ell = +1 \mid I)}{\pi(\ell = -1 \mid I)}\right)$$

More examples on detection


$$K_{mr}(x,y) = \min\left(1, \frac{q_{ml}(x \mid y;\ \mathrm{F(I)})}{q_{mr}(y \mid x;\ \mathrm{F(I)})} \cdot \frac{\pi(y \mid \mathrm{I})}{\pi(x \mid \mathrm{I})}\right)$$

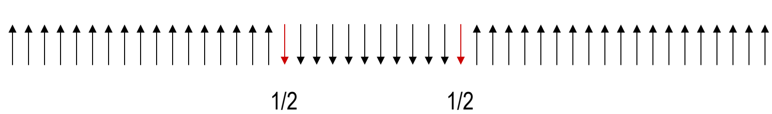
It is the ratio of the discriminative proposal probabilities rectified by the ratio of the generative probabilities. When the discriminative models are good approximations the acceptance is close to one. Then the MC becomes very effective.

Struggles of Gibbs sampler with Ising / Potts models

$$p(I) = \frac{1}{Z}\exp\{-\beta \sum_{<s,t>} 1(\mathrm{I}_s = \mathrm{I}_t)\} = \frac{1}{Z}\prod_{<s,t>} \exp\{-\beta \cdot 1(\mathrm{I}_s = \mathrm{I}_t)\}, \quad <s,t> \in E_o$$

For example, in a 1D string of spins, suppose we use a Gibbs sampler to flip one spin at a time It has a p=½ probability for flipping the spin at the boundary. Flipping a string of length n will need on average

$t = 1/p^n = 2^n$ steps!

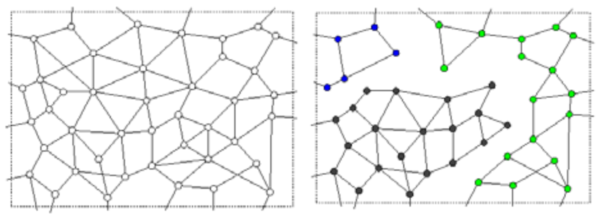


So the Gibbs sampler experiences exponential waiting time.

Sampling in the graph partition space

There are two types of subspaces: object spaces + partition spaces

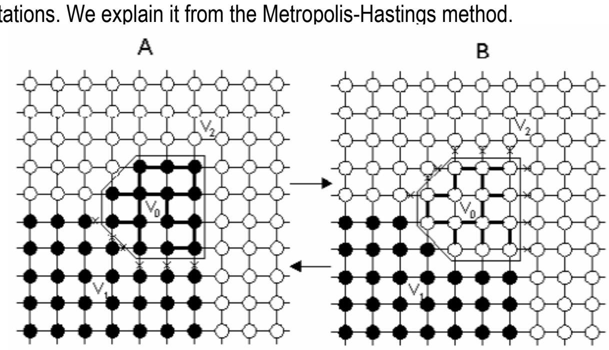
The partition space is the set of all possible partition (coloring) of a graph (e.g. pixel lattice, edge maps, …)



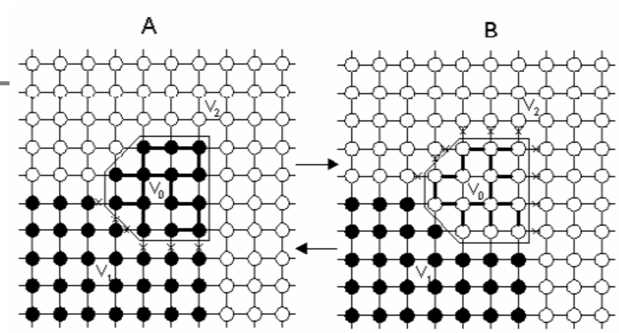
How do we sample a general posterior probability in the partition space?

Swendsen-Wang with Ising / Potts models

Swedsen-Wang (1987) is an extremely smart idea that flips a patch at a time. There are multiple interpretations. We explain it from the Metropolis-Hastings method.



Each edge in the lattice e=<s,t> is associated with a constant probability q.
If s and t have different labels at the current state, e is turned off.
If s and t have the same label, e is turned off with probability q.
Thus each object is broken into a number of connected components (subgraph).



Consider the reversible moves between states A and B by Metroplis-Hastings:

the proposal probability ratio is:

$$\frac{q(A \to B)}{q(B \to A)} = \frac{(1-q_o)^{|C(V_0,V_1)|}}{(1-q_o)^{|C(V_0,V_2)|}} = (1-q_o)^{|C(V_0,V_1)|-|C(V_0,V_2)|}$$

the probability ratio of the two states is:

$$\frac{p(A)}{p(B)} = \frac{\exp^{-\beta \cdot |C(V_0,V_2)|}}{\exp^{-\beta \cdot |C(V_0,V_1)|}} = \exp^{\beta \cdot (|C(V_0,V_1)|-|C(V_0,V_2)|)}$$

SW with Ising / Potts models

The acceptance probability for the move from A to B is,

$$\boxed{\alpha(A \to B) = \min(1, \frac{q(B \to A)}{q(A \to B)} \cdot \frac{p(B)}{p(A)}) = (\frac{e^{-\beta}}{1-q_o})^{|C(V_o,V_1)|-|C(V_o,V_2)|} = \frac{e^{-\beta}}{1-q_o}}$$

If we choose

$$\boxed{q_o = 1 - e^{-\beta}, \qquad \beta \propto \tfrac{1}{T}}$$

Then the acceptance probability is always 1.

At very high temperature, q_o is close to zero, the SW is reduced to Gibbs sampler.
At very low temperature, q_o is close to one, the SW can flip very large patch at each step.

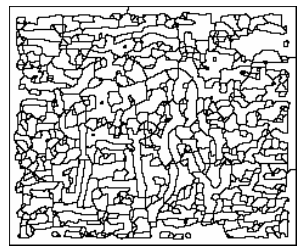
Recently it is proven that SW mixes in polynomial time (Cooper and Frieze).

The Partition space for image segmentation

The Swendsen-Wang method was limited to Ising/Potts models on lattice, we generalized it to sampling general posterior probabilities on general graphs

(Barbu and Zhu, ICCV03)

input image



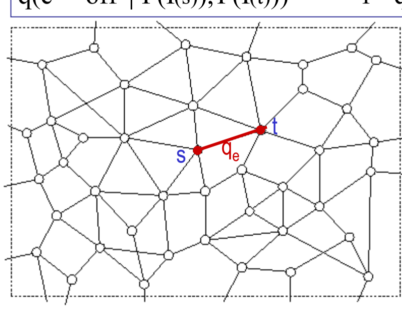
over-segmentation
with atomic regions

Walking in the Partition Space

An probabilistic adjacency graph: G_o=<V, E_o, Q>

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

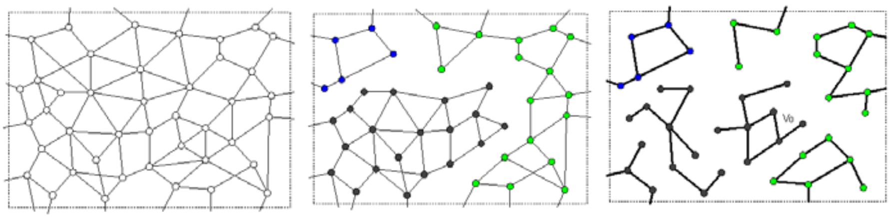
$$\boxed{\frac{q(e=\text{"on"} \mid F(I(s)), F(I(t)))}{q(e=\text{"off"} \mid F(I(s)), F(I(t)))} = \frac{q_e}{1-q_e}}$$



q_e can be obtained by supervised learning:

1). Konishi, Yuille et al 01
2). Adaboost by Shapire 00
3). Geisler, 00.
… …

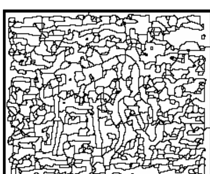
Graph clustering



a. adjacency graph

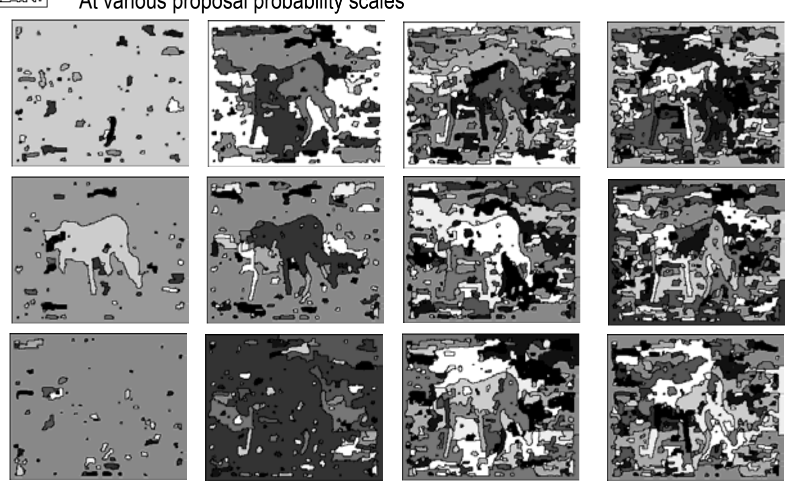
b. current partition and labeling
edges between different colors are removed

c. Graph clustering
dark edges are turned “on”, other edges are turned “off”. We get some connected sub-graphs



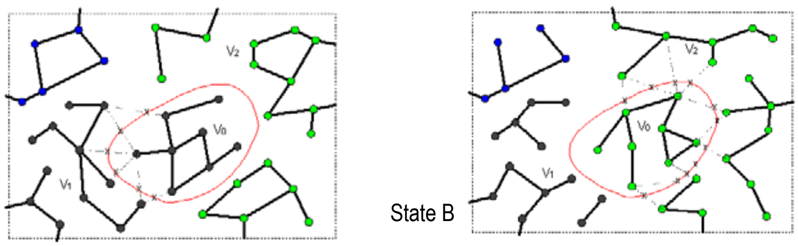
Graph partition/clustering:
sampling the discriminative model in the partition space

At various proposal probability scales



Markov chain moves: Flipping a sub-graph

State A



State B

$$\alpha(A \to B) = \min(1, \frac{q(B \to A)}{q(A \to B)} \cdot \frac{p(B)}{p(A)})$$
$$= \min(1, \frac{\Pi_{e \in C(V_0, V_2)}(1-q_e)}{\Pi_{e \in C(V_0, V_1)}(1-q_e)} \cdot \frac{p(l_1 | V_0)}{p(l_2 | V_0)} \cdot \frac{p(V_2)}{p(V_1)})$$

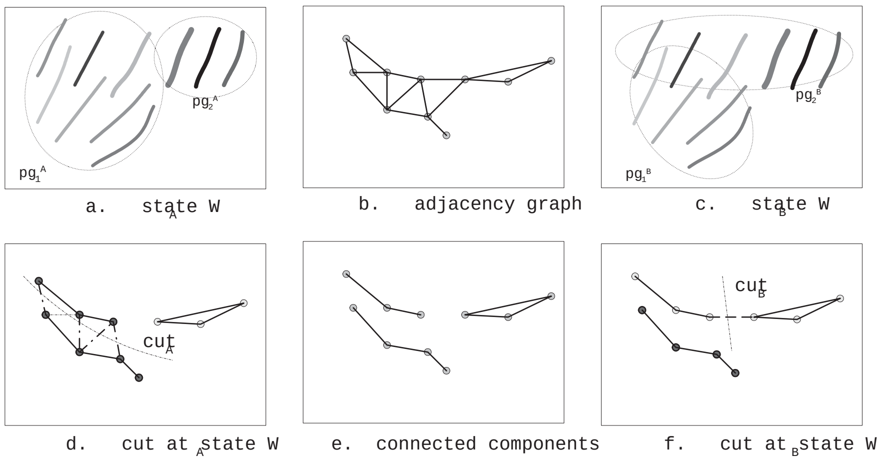
Markov chain moves: Flipping a sub-graph

$$\alpha(A \to B) = \min(1, \frac{\Pi_{e \in C(V_0, V_2)}(1-q_e)}{\Pi_{e \in C(V_0, V_1)}(1-q_e)} \cdot \frac{p(l_1 | V_0)}{p(l_2 | V_0)} \cdot \frac{p(B)}{p(A)}) = 1$$

If we select the label probability as

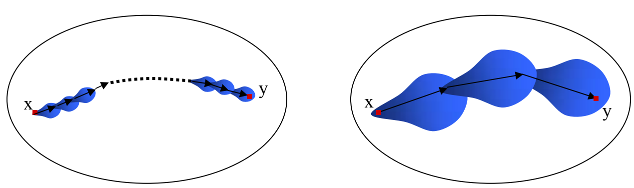
$$p(l_1 | V_0) = \Pi_{e \in C(V_0, V_1)}(1-q_e) \cdot p(V_1)$$
$$p(l_2 | V_0) = \Pi_{e \in C(V_0, V_2)}(1-q_e) \cdot p(V_2)$$

Another example on partition: Curve grouping



Sampling the partition spaces by Swendsen-Wang Cut

The basic idea is to enlarge the jump scopes by designing Composite Jumps



Searching with a torch vs Searching with a long range RADAR

Computational cost / complexity

1. Markov chain mixing time
   ----- large scopes yields transition kernel K(x,y) with large conductance and thus fast mixing.

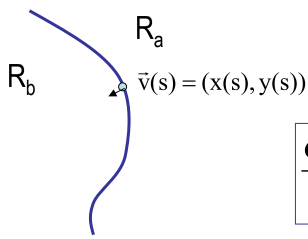
2. Cost of computing proposal probabilities in a large scope.

The Diffusion Components by PDEs

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.

the region competition for curve evolution (Zhu and Yuille, 96)



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

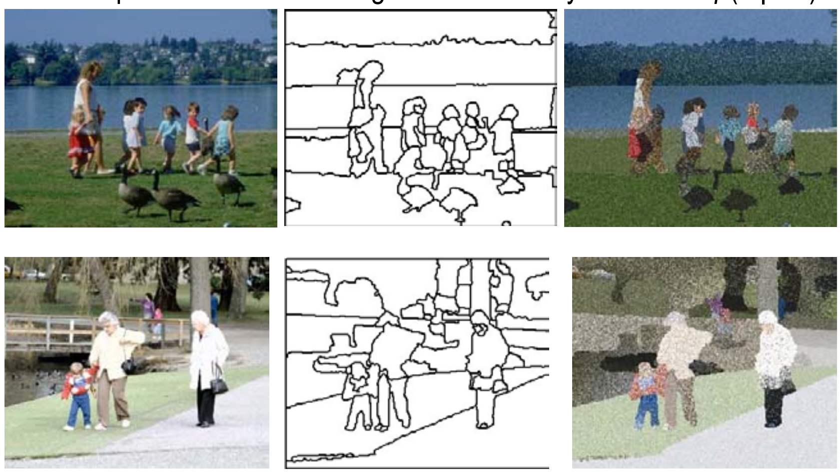
$$\boxed{\frac{d\vec{v}(s)}{dt} = (\mu \cdot \kappa(s) + \frac{\log p(I(x,y)|\ \theta_a)}{\log p(I(x,y)|\ \theta_b)}\ ) \cdot \vec{n}(s)}$$

Experiment I: Color Image Segmentation

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

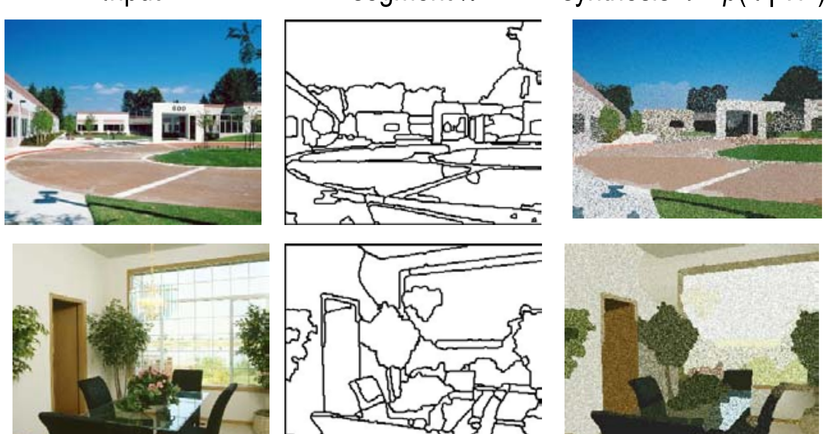
Experiment I: Color Image Segmentation

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

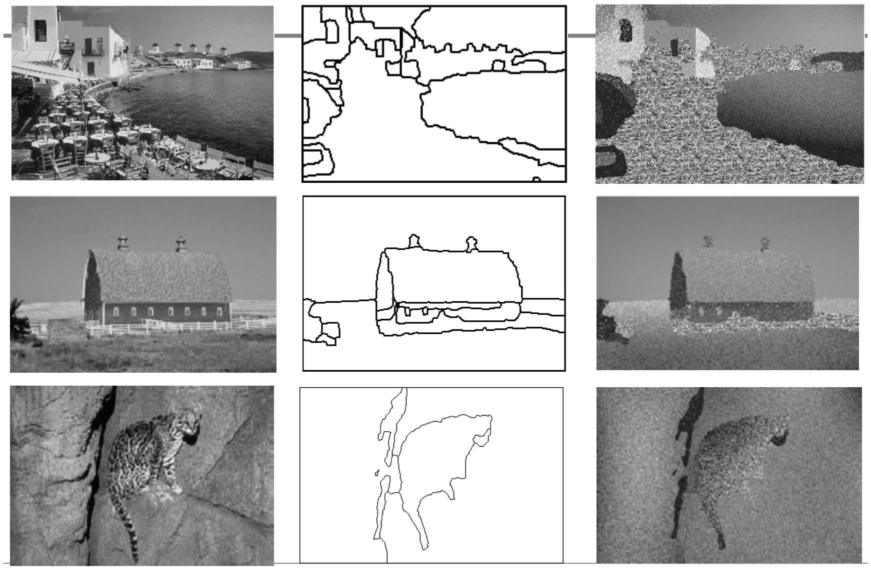


Experiment I: Color Image Segmentation

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

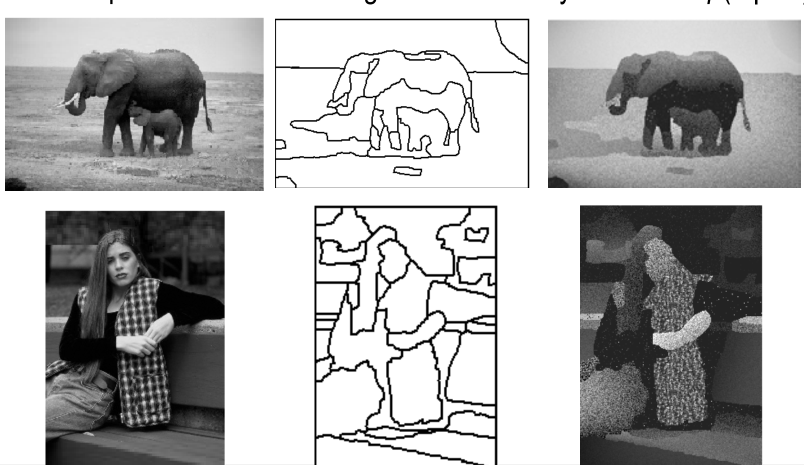


a. Input image b. segmented regions c. synthesis $I \sim p(I \mid W^*)$



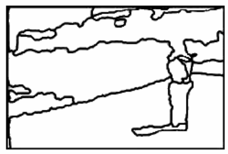
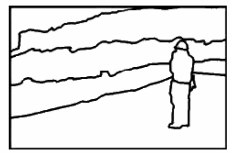
Grey Level Image Segmentation

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

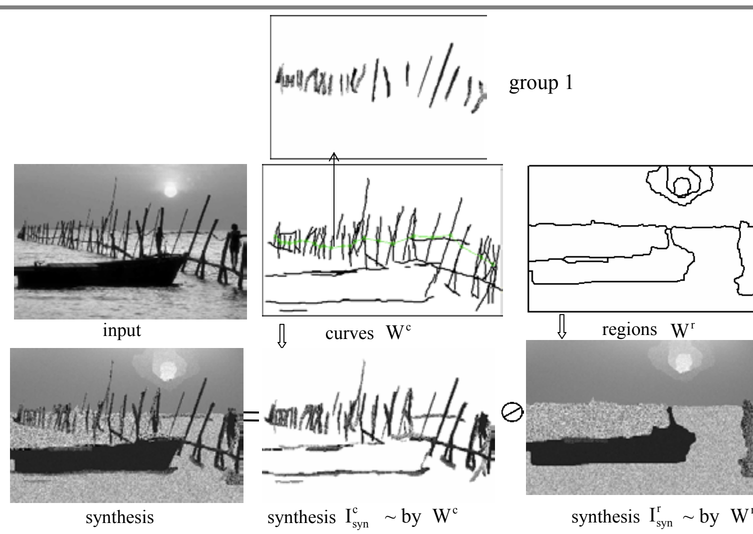


The Berkeley Benchmark Study

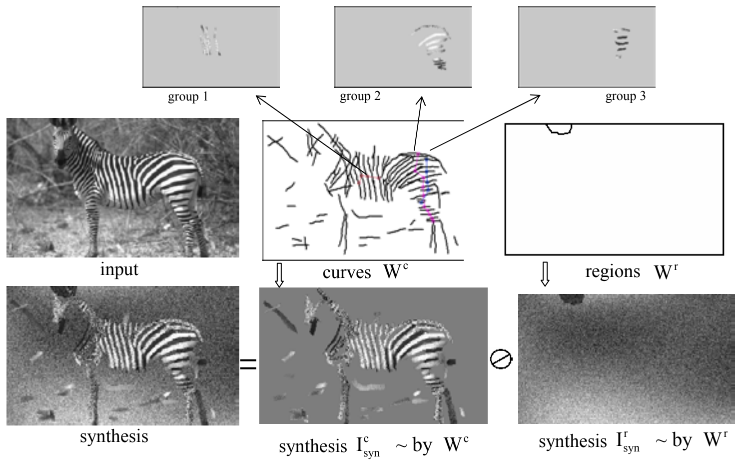
(David Martin, 2001)

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

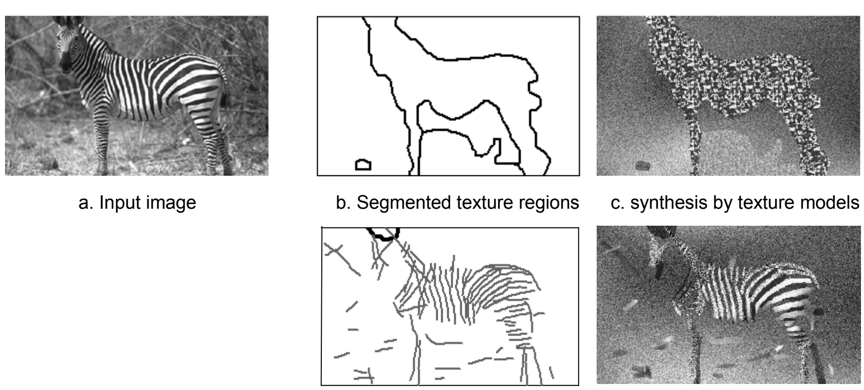
Experiment II: Regions + Curves



Experiment II: region + curves



Computing Ambiguity in Visual Inference



a. Input image

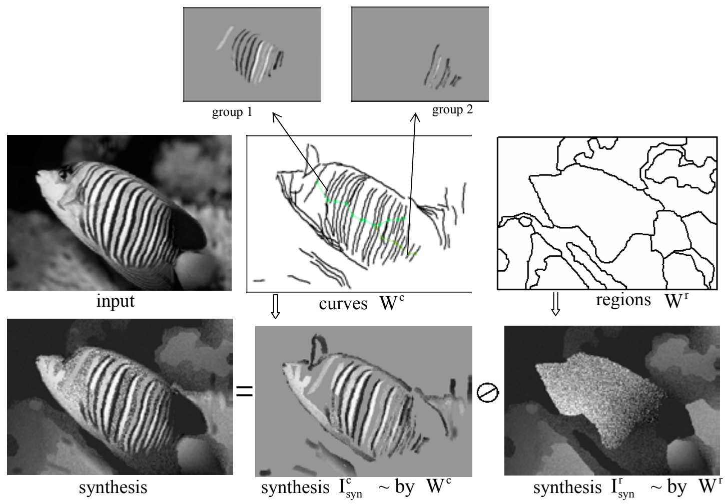
b. Segmented texture regions

c. synthesis by texture models

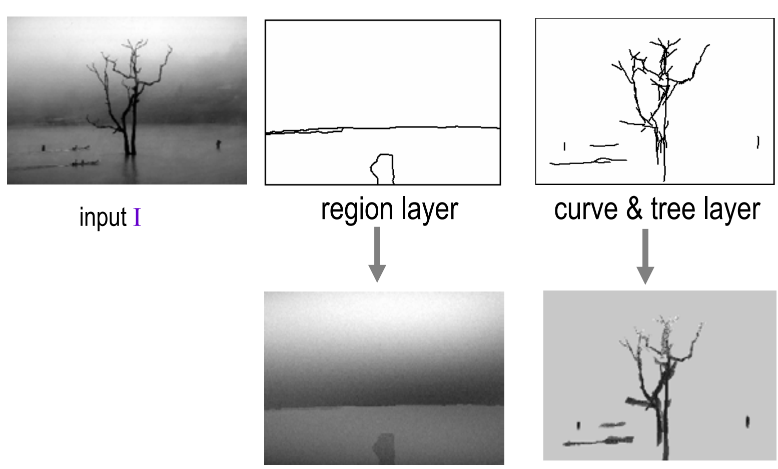
d. curve processes + bkgd region

e. synthesis by curve models

Experiment II: Regions + Curves



Experiment III: Regions+ curve



Example IV: 3D reconstruction from a single image

Solution 1:

(Han and Zhu, 2003)

Solution 2:

Experiment IV: Regions+ curve + 3D reconstruction

Example on 3D reconstruction (Han and Zhu, 2003)

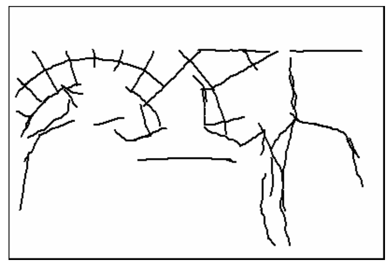
input image

3D reconstruction

Input image

Synthesized image

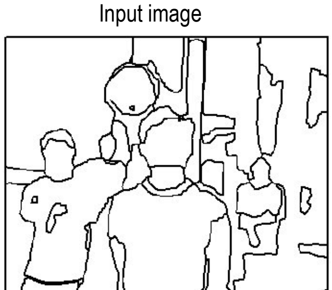
Faces, words, and regions



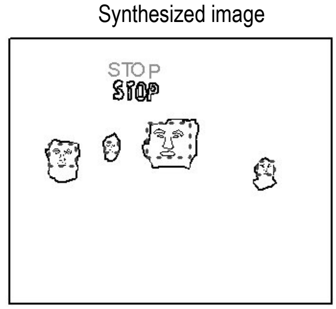
Curves

Input image

Synthesized image

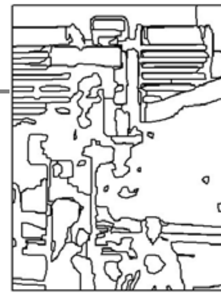


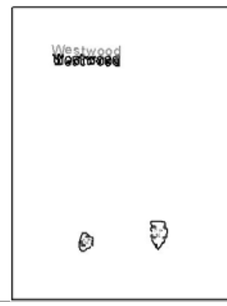
Faces, words, and regions

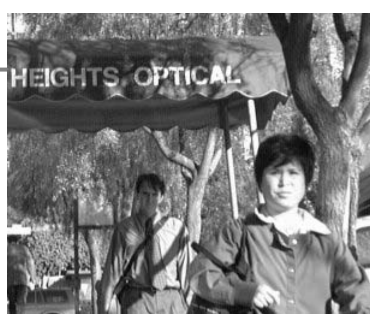


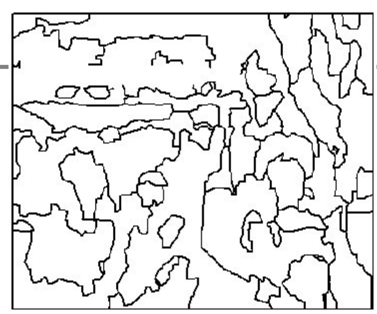
Curves

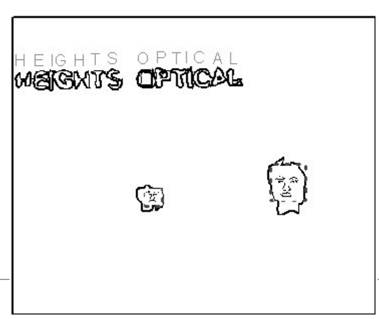

Westwood




Westwood
Westwood

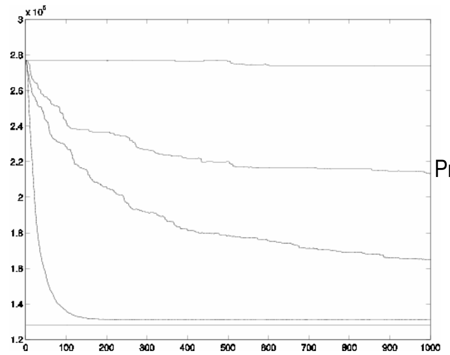

HEIGHTS OPTICAL




HEIGHTS OPTICAL
HEIGHTS OPTICAL

Empirical Speed Comparison 1

Comparison in the object spaces



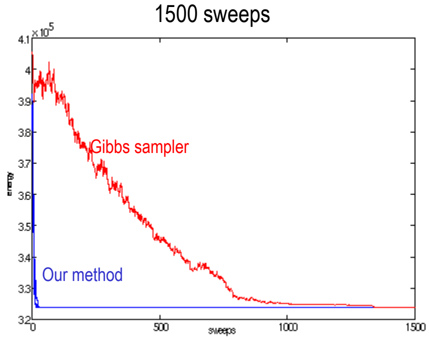
Proposals by uniform prob.

Proposals by discriminative prob.

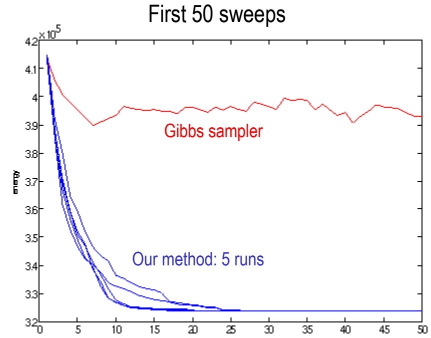
Empirical Speed Comparison 2

In the partition space: SW-cut vs Gibbs sampler.

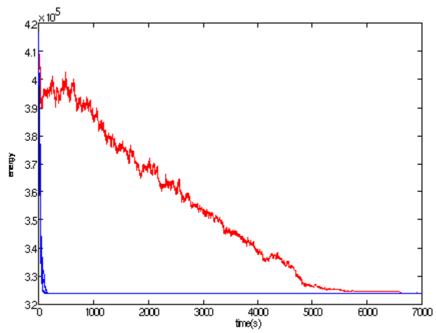
1500 sweeps



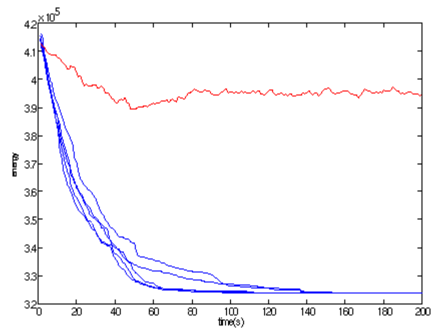
First 50 sweeps



Empirical speed comparison: in seconds



7000 seconds



zoom-in view of the first 200 seconds

Theoretical Speed Analysis I

A direct speed measure is the *first-hitting time*:

It is the number of steps for reaching a state x for the first time

$$\tau_{hit}(x) = \min\{\, n \geq 1 : x_n = x \,\}$$

Theorem:

The expected first hitting time for a MC=(ν, K, π) is

$$\boxed{E[\tau_{mix}(x)] = 1 + \nu'_{-x}(I - K_{-x})^{-1}1}$$

where 1=(1,1,….,1), and –x means the row and column for x are removed.

Theoretical Speed Analysis I

We can compute the first-hitting time explicitly for some simple case

(Maciuca and Zhu, 2003)

Theorem

Consider sampling a target probability p(x) with a proposal probability q(x) on space W. For any state x, we have

$$\boxed{\frac{1}{\min(p(x),q(x))} \le E[\tau_{hit}(x)] = \frac{1}{p(x)(1-\lambda(x))} \le \frac{1}{\min(p(x),q(x))} \cdot \frac{1}{1-d_{TV}(p,q)}}$$

where

$$d_{TV}(p,q) = \frac{1}{2}\sum_{x\in\Omega} \| p(x) - q(x) \|$$

Theoretical Analysis II

The 2nd speed measure is the *mixing time*:

A Markov chain starts at state x0, and after t=n steps, it follows a probability $K^t(x_0, x)$, it is apart from the target probability p by a total variance distance

$$d_{TV}(t) = \frac{1}{2}\sum_{x\in\Omega} \| K^t(x_0, x) - p(x) \|$$

The Markov chain mixing time is defined as

$$\tau_{mix}(\varepsilon) = \max_{x_o\in\Omega} \min_t \{ d_{TV}(t) < \varepsilon \}$$

Theorem

$$d_{TV}(t) \le \sqrt{\frac{1-p(x_0)}{4p(x_0)}} \quad \lambda^t_{slem}$$

$0 < \lambda_{slem} < 1$ Is the second largest eigen value modulus of the transition matrix K.

Computing Strategies

1. Generative methods --- "Top-down",
   --Markov chain Monte Carlo for jumps
   -- PDEs for diffusion.

   compute the posterior prob. ratios with full generative model over the entire image.

2. Discriminative methods --- "Bottom-up"
   -- Data clustering, Adaboost, ….

   compute the posterior prob. ratios in a factorized form within local image in separate vision cues/channels.

Discussion 1: Computing Strategies

We consider each bottom-up and top-down step as a "test"
Generative test: Accurate but slow
Discriminative test: Fast but biased

What is the best strategy for ordering these tests?

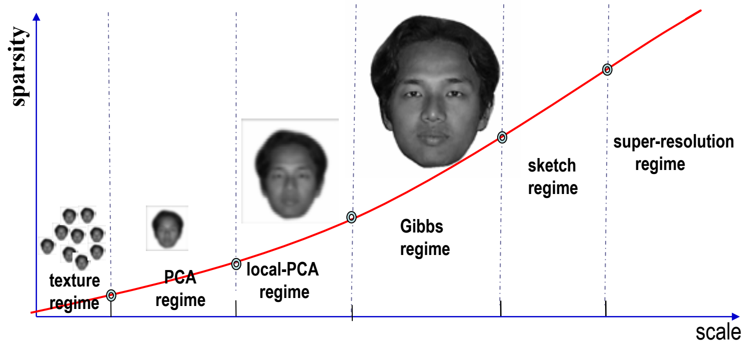
(Blanchard and D.Geman 03)

A test is characterized by its *POWER* and *COST*. Under some simple case, they showed that the best strategy is to order the test according to their power/cost ratio.

Different data ensemble may need different strategies ---"*pathways*"

Discussion 2: What is the structure of Object Space?

From coarse-to-fine scales, our perceptual models must experience quantum jumps over a serious of probability families.



How do we augment the probability models?

References

1. S. C. Zhu, R. Zhang, and Z. W. Tu "*Integrating Top-down/Bottom-up for Object Recognition by DDMCMC*", CVPR 2000
2. Z.W. Tu and S.C. Zhu, "*Image Segmentation by DDMCMC*", PAMI 2002.
3. Z.W. Tu and S.C. Zhu, "*Parsing images into region and curve processes*", ECCV 2002
4. Z.W. Tu, X.R. Chen, A.L Yuille, and S.C. Zhu. "*Image parsing: segmentation, detection, and recognition*", ICCV 2003.
5. A. Barbu and S.C. Zhu, "*Graph partition by Swendsen-Wang cut*", ICCV 2003.
6. R. Maciuca and S.C. Zhu, "*How do heuristics expedite Markov chain search*?" SCTV 2003