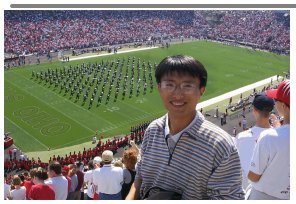
Visual Inference by Data-Driven Markov Chain Monte Carlo

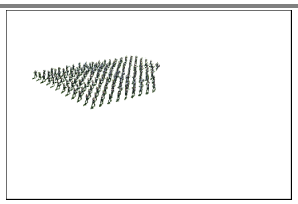
Zhuowen Tu and Song-Chun Zhu

Statistics and Computer Science
University of California, Los Angeles

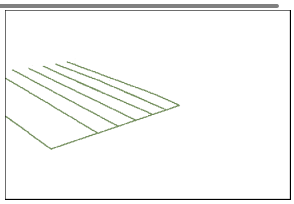
Parsing Image Into Various Stochastic Patterns



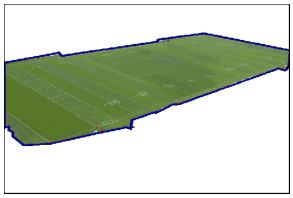
input image



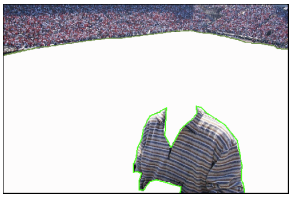
point process



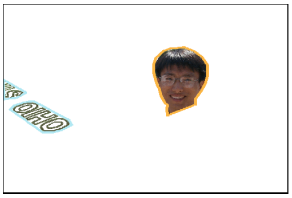
curve process



a color region



texture regions



objects

Depending on the types of patterns it focuses, image parsing subsumes conventional vision tasks:
perceptual organization, image segmentation, object recognition, etc.

A Bayesian Formulation

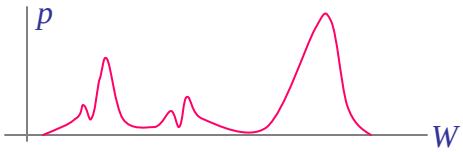
A basic assumption, dated back to Helmholtz (1860), is that biologic and machine vision is to compute the most probable interpretation(s) from input images.

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

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

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

$$(W_1, W_2, \ldots, W_k) \sim p(W \mid I)$$



Problems

1. Representational or modeling problems:

   What are W , p(W), and p(I | W) ?

2. Computational problems:

   a). What are the structures of the search space, which we call Ω?

   b). How do we explore the search space for globally optimal solutions ?

   --- reversible MC jumps + diffusion (PDEs).

   c). How do we compute and preserve ambiguities .

Can MCMC run in seconds on a PC for parsing images?

Ideas to Improve MCMC Speed in Literature

A main idea is to introduce auxiliary random variables:

$$x \sim \pi(x)$$

Augment x by variables:

T	--- temperature	(Simulated tempering, Narinari and Parisi, 92, Geyer and Thompson, 95)
s	--- scale	(Multi-grid sampling, Goodman and Sokal 88, Liu et al 94)
w	--- weight	(dynamic weighting, Liang and Wong 1996)
b	--- bond	(clustering, Swendsen-Wang, 87)
u	--- energy level	(slice sampling, Edwards and Sokal, 88 ...)

The common problem is:
The Markov chain moves are designed a priori, without looking at the data.

What is Data-Driven Markov Chain Monte Carlo ?

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

Consider a reversible jump $W_A \leftrightarrow W_B$

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

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

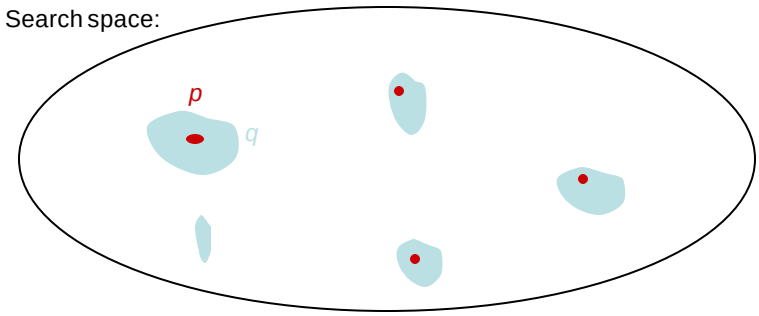
In DDMCMC,

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

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

Then it may converges in a small number of steps !

Basic Ideas



The proposal probabilities q( ) focuses on a tiny portion of the search space and thus narrows the search exponentially in a probabilistic fashion. Thus the Markov chain converges and mixes very fast.

Intuitive Idea: Divide-and-Conquer

Let $W=(w_1, w_2, \ldots, w_n)$, usually these variables are divided for several types:

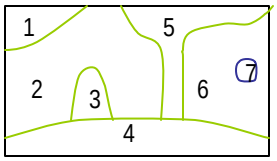
partition, label of models, model parameters, order, …

Consequently, the search space is made of a few types of "atomic spaces" --- one for each type of variables --- through union and production.

Then we can compute discriminative probabilities in each atomic space, which is then composed into the proposal probabilities.

Example: Image Segmentation

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



$p_7 = (R_1, R_2, \ldots, R_7)$ is a 7-partition of the lattice.

$$\nu_{p_n} = \{ p_n = (R_1, R_2, \ldots, R_n) : \bigcup_{i=1}^{n} R_i = \Lambda, R_i \cap R_j = \varnothing, \forall i \neq j \} \; / \; pg$$

The *partition space* is $\nu_p = \bigcup_{n=1}^{|\Lambda|} \nu_{p_n}$

A permutation group

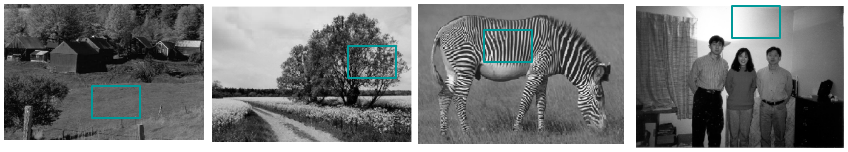
Some Image Models

Some families of image models:

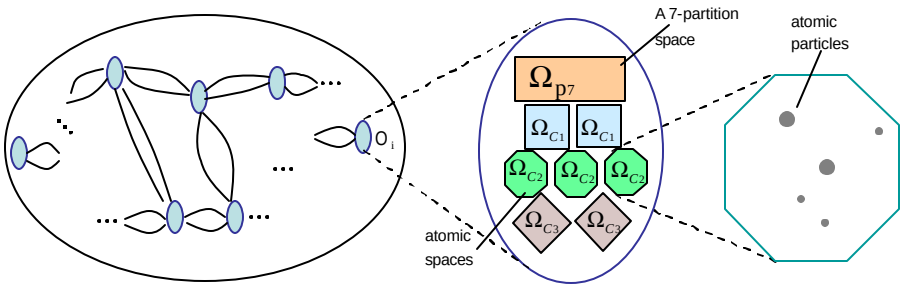
- ν_{g1} : iid Gaussian for pixel intensities
- ν_{g2} : non-parametric histograms
- ν_{g3} : Markov random fields for texture
- ν_{g4} : Spline model for lighting variations

- ν_{c1} : iid Gaussian for color (LUV)
- ν_{c2} : mixture of Gaussians for color
- ν_{c3} : spline model for smooth color variations (e.g. sky, lake, ...)

ν_{g1} ν_{g2} ν_{g3} ν_{g4}



The Search Space



a). solution space b). a sub-space of 7 regions c). an atomic space

Designing Markov Chain Dynamics

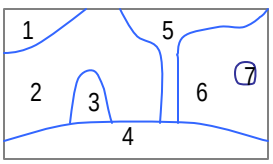
Type I: Diffusion of region boundary -- region competition.

Type II: Splitting of a region into two.

Type III: Merging two regions into one.

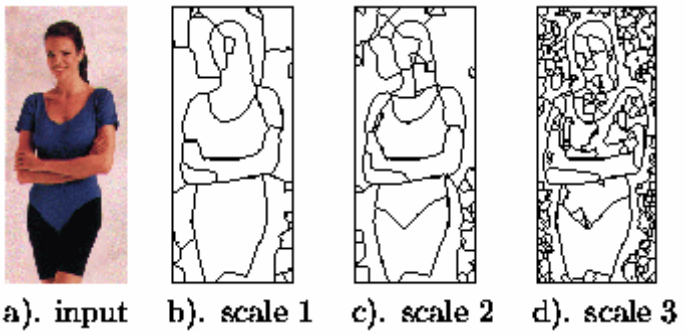
Type IV: Switching the family of models for a region.

Type V: Model adaptation for a region.



Edges in Partition Space v_p

Edge detection and tracing at three scales of details:

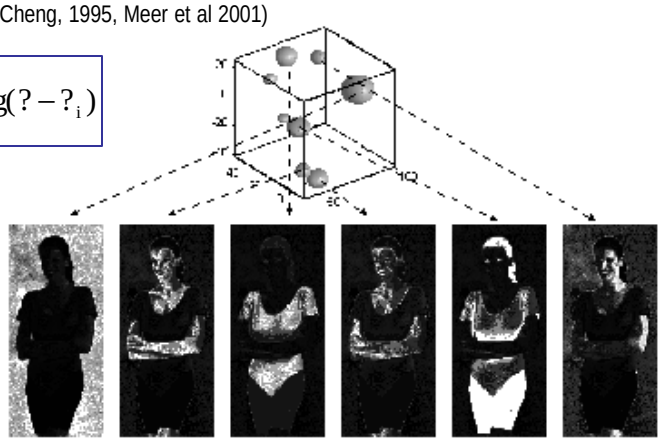


Clustering in Color Space v_{c1}

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

$$q(?|I) = \sum_{i=1}^{K} ?_i \, g(? - ?_i)$$

Input



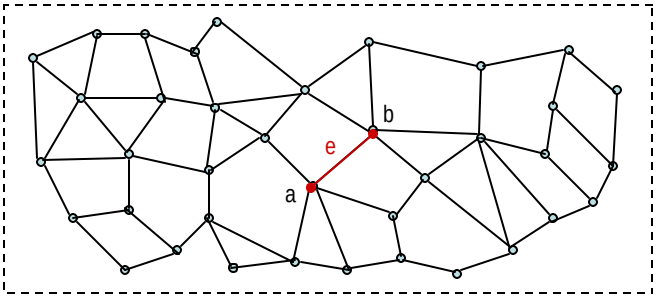
saliency maps 1 2 3 4 5 6

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

Walking in the Partition Space

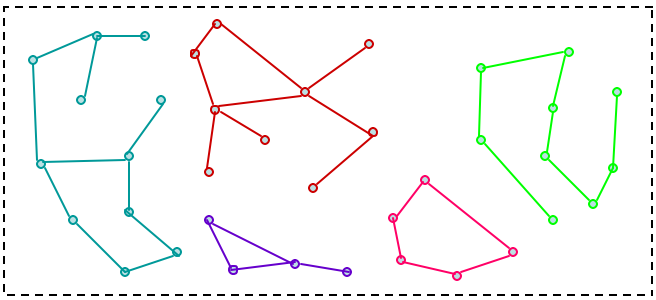
an adjacency graph: each vertex is a basic element : pixels, small-regions, edges, ….
each link e=<a, b> is associated with a probability/ratio for similarity

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



Walking in the Partition Space

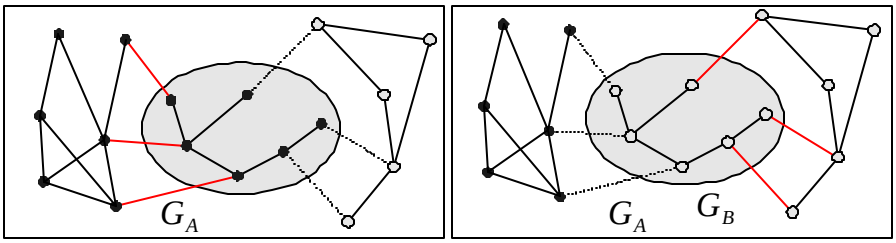
Sampling the edges independently, we get connected components:



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

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

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:

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

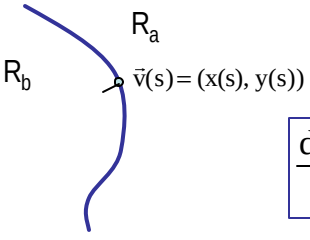
results in an ergodic and reversible Markov Chain.

Diffusion 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.

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

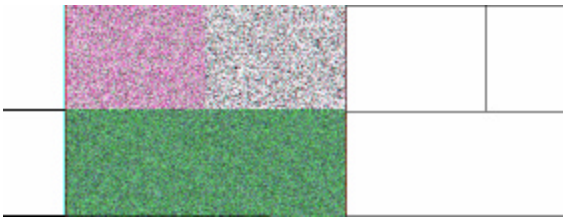


Let v be a point on the boundary between two regions, its motion is governed by the region-competition equation.

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

Results by DDMCMC

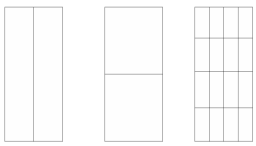
segmentation synthesis



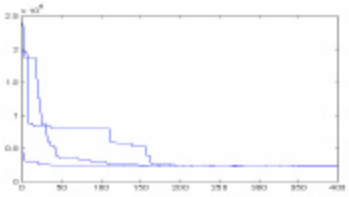
snapshot of solution W sampled by DDMCMC

Running DDMCMC

starting with 3 different initial segments below



MC 1 MC 2 MC 3

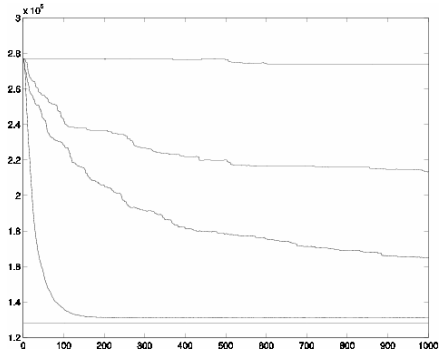


energy plots of three MCMCs

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

Performance Comparison

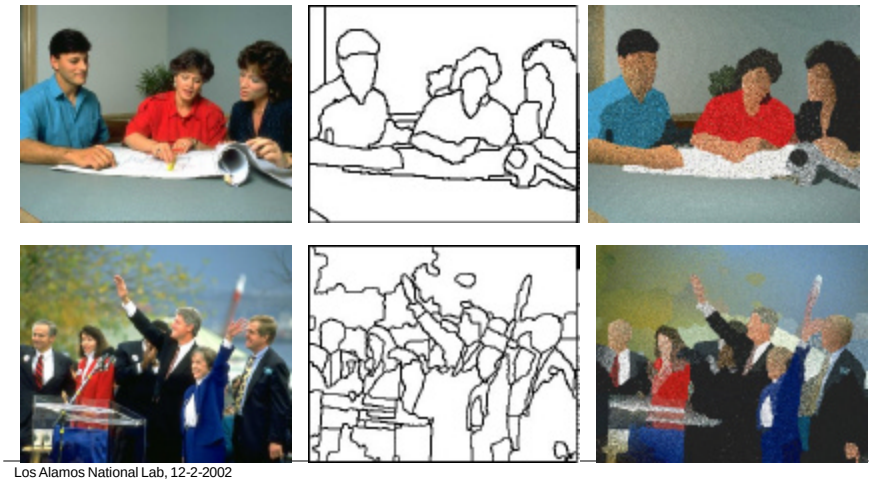
Analyze performance bounds of DDMCMC paradigm.



DDMCMC are 2-3 orders of magnitude faster than traditional MCMC.

Experiments: Color Image Segmentation

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



Experiments: Color Image Segmentation

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

Los Alamos National Lab, 12-2-2002

Experiments: Color Image Segmentation

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

Los Alamos National Lab, 12-2-2002

a. Input image b. segmented regions c. synthesis I ~ p(I | W*)

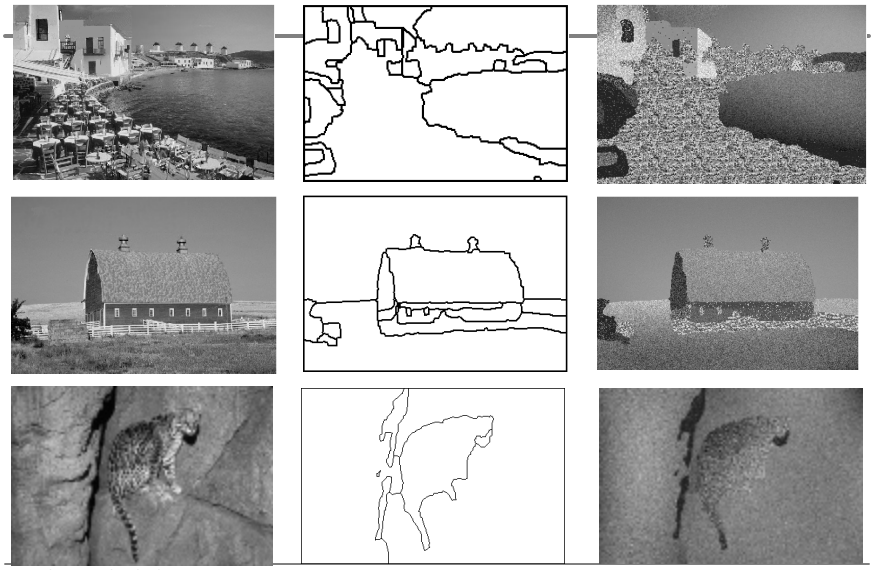
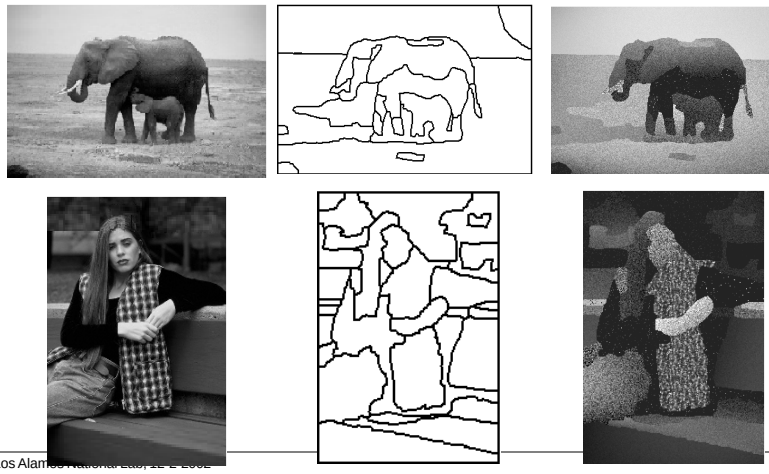


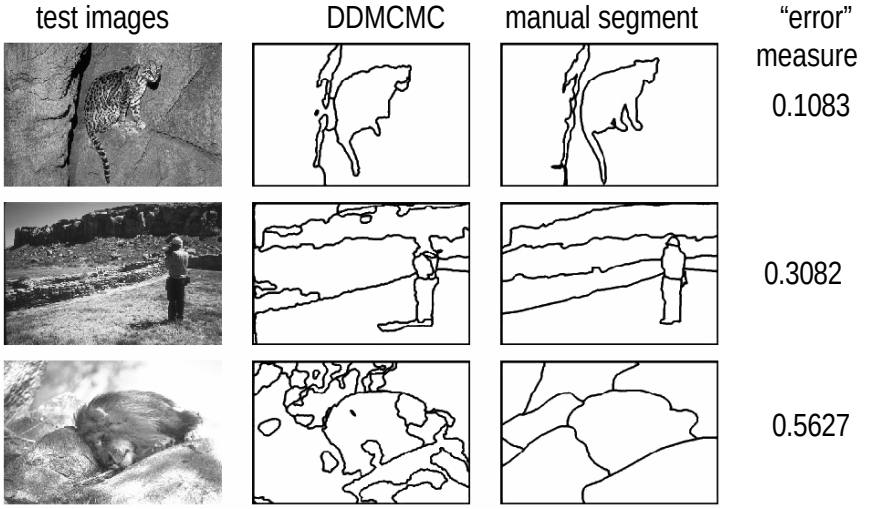
Image Segmentation

Input segment π^* synthesis I ~ p(I | W*)

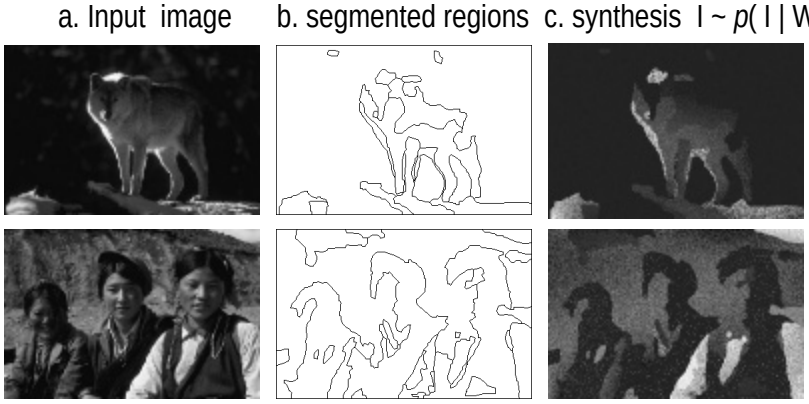


The Berkeley Benchmark Study

(David Martin et al, 2001)

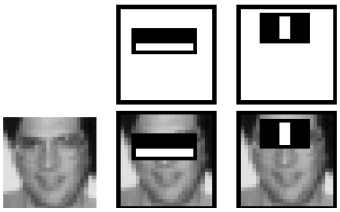


Examples of Failure

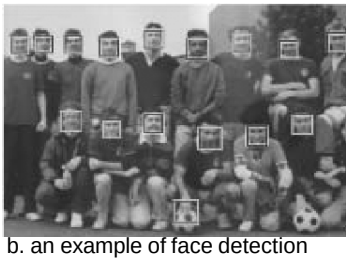


Adaboost in the Label Space

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



a. the first two face features

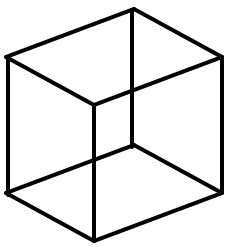


b. an example of face detection

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

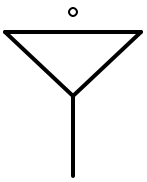
$$y=\text{Sign}(a_1 h_1(I) + \ldots + a_T h_T (I)) \rightarrow \text{sign}(p(y=1|\, I)/\, p(y=-1|\, I)$$

Ambiguities in Visual Inference



Nicker cube

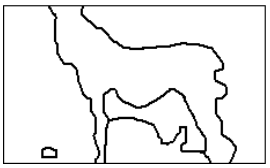
Vase vs. faces



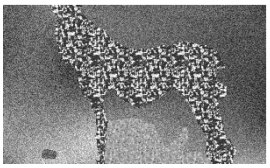
bikini vs. martini

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

Computing Multiple Solutions

To faithfully preserve the posterior probability $p(W|\mathbf{I})$,

We compute *a set of weighted scene particles* $\{W_1, W_2, \ldots, W_M\}$,

$$\hat{p}(W|\mathbf{I}) = \sum_{i=1}^{M} \alpha_i G(W - W_i), \quad \sum_{i=1}^{M} \alpha_i = 1$$

A mathematical principle:

$$S^* = \{W_1, W_2, ..., W_M\} = \arg\min_S D(p||\hat{p})$$

Pursuit of Multiple Solutions

$$D(p||\hat{p}) \approx \log\frac{\varpi}{\boldsymbol{\omega}} + \sum_{n=1}^{N}\frac{\omega_n}{\boldsymbol{\omega}}[(E(\mathbf{x}_{\tau(c(n))}) - E(\mathbf{x}_n)) + \frac{(\mathbf{x}_n - \mathbf{x}_{\tau(c(n))})^2}{2\sigma^2}]$$
$$= \hat{D}(p||\hat{p})$$

The Kullback-Leibler divergence can be computed if we assume mixture of Gaussian distributions.

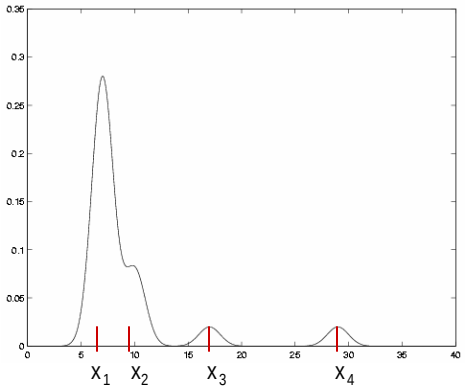
--- a simple fact: the KL-divergence of two Gaussians is the signal-to-noise ratio.

Intuition: S includes global maximum, local modes, apart from each other.

A *k*-adventurer algorithm

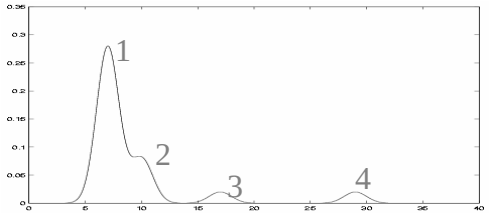
1. Initializing S_k by one $\mathbf{x}$ repeated k times and initializing $\hat{p}(\mathbf{x})$.
2. Repeat
3. Get k new weighted particle $(\omega_i', \mathbf{x}_i')$ by k MCMCs starting at $\mathbf{x}_i$ respectively.
4. $S_+ \leftarrow S_k \cup \{(\omega_i', \mathbf{x}_i'), i = 1, ..., k\}$.
5. $p(\mathbf{x}) \leftarrow \hat{p}(\mathbf{x})$ by adding new particles.
6. $s^* = \arg\min_{|s|=k, s\in S_+} D(p||\hat{p}_+(s))$.
7. $S_k \leftarrow s^*$.

Preserving Distinct Particles



An Example of Keeping Multiple Solutions

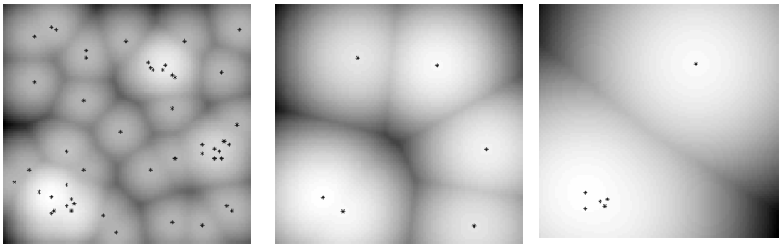
An example of illustration:



chosen S_3:	$\{x_1, x_2, x_3\}$	$\{x_1, x_2, x_4\}$	$\{x_1, x_3, x_4\}$	$\{x_2, x_3, x_4\}$
$D(p\|\|\hat{p})$:	3.5487	1.1029	0.5373	2.9430
$\hat{D}(p\|\|\hat{p})$:	3.5487	1.1044	0.4263	2.8230
$\|p - \hat{p}\|$:	0.1000	0.1000	0.3500	1.2482

An example of illustration:

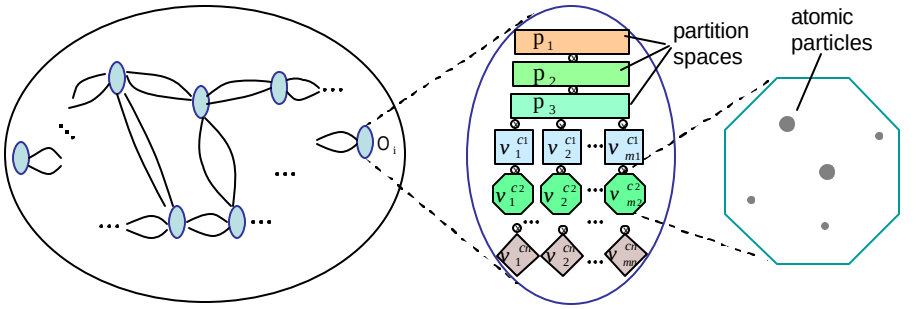
$\min D(q \| q_1)$ $\min | q - q_2 |$



q q_1 q_2

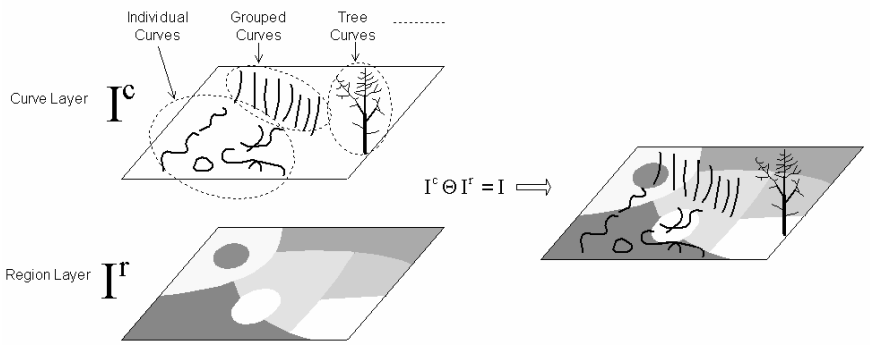
A model p with 50 particles two approximate models q with 6 particles

General Search Space for Image Parsing



a). solution space Ω b). a sub-space Ω_i c). an atomic space v_{m2}^{c2}

Parsing Images into Regions and Curves

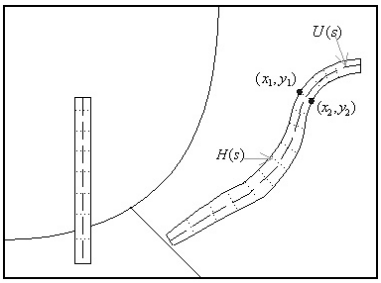


$$W = (W^r, W^c, W^p, W^t)$$

$$W^c = (K^c, \{C_i; i = 1, ..., K^c\}, N^p, \{P_i; i = 1, ..., K^p\}, N^t, \{T_i; i = 1, ..., K^t\})$$

Curve Models

$$W^c = (K^c, \{(C_i, a_i); i = 1..K^c\})$$



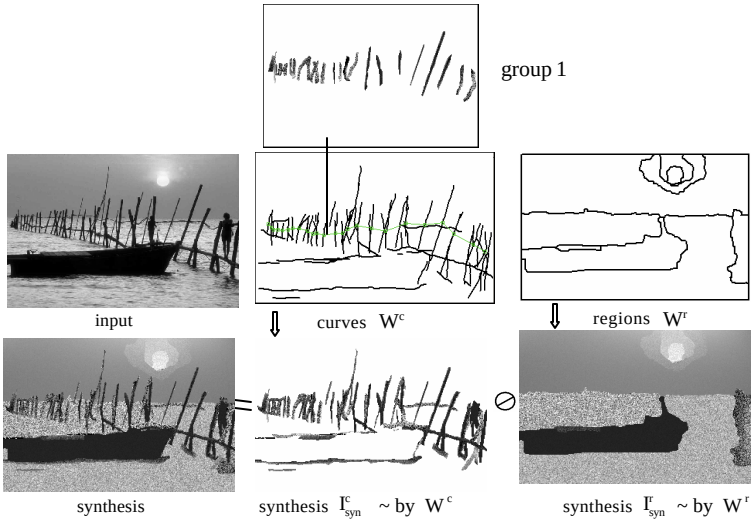
Curve $C = (\Gamma, q)$

Curve shape $\Gamma = (U(s), H(s))$

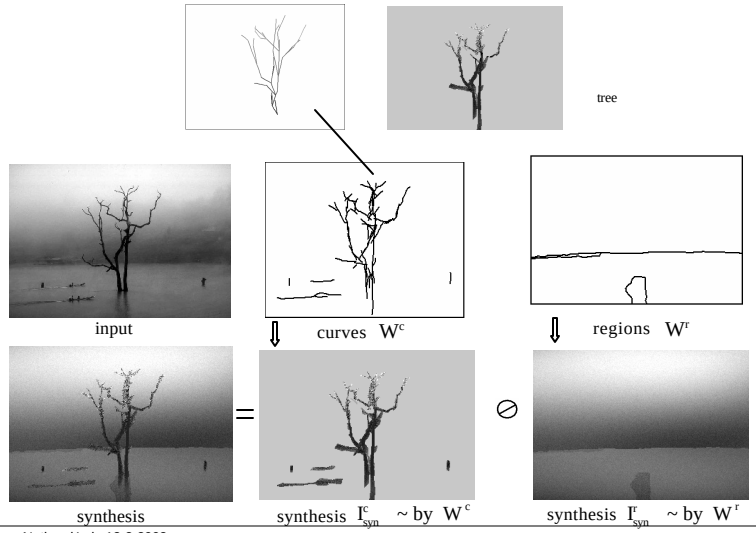
$H(s)$ is the curve width.

$U(s)$ is the center line.

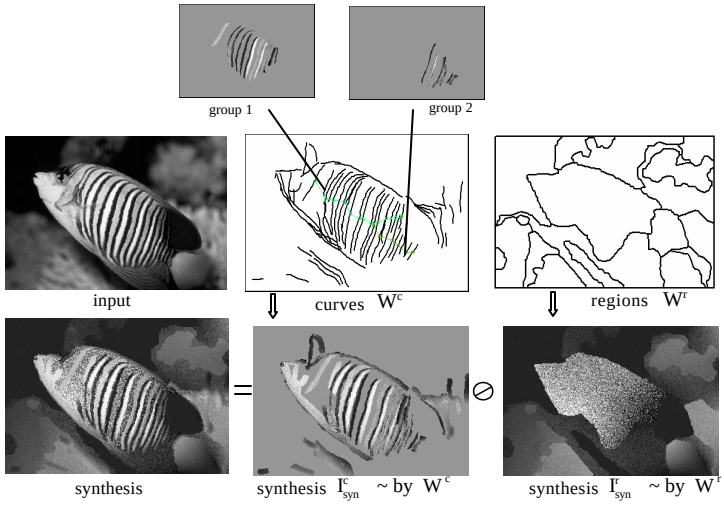
Parse Image into Regions and Curves



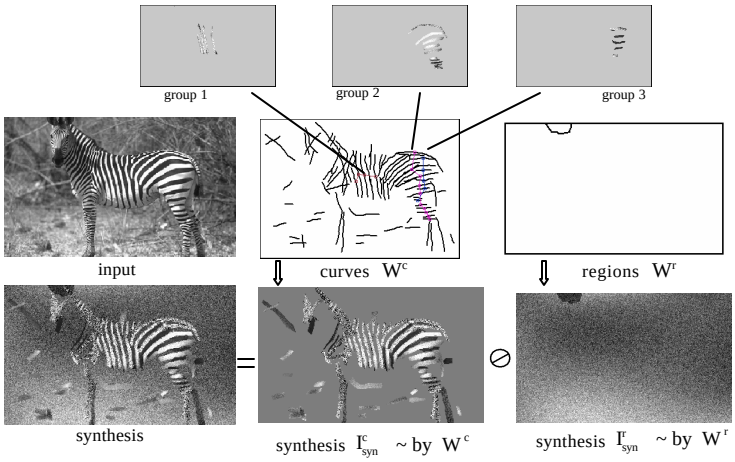
Parsing Images with Trees



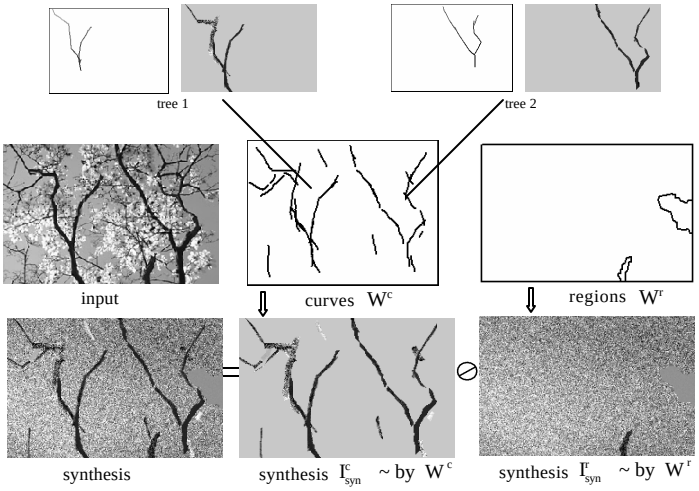
Parse Image into Regions and Curves



Parse Image into Regions and Curves



Parsing Images with Trees

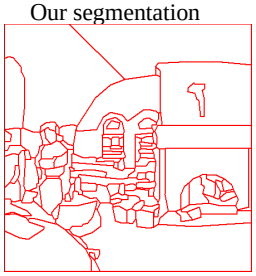


Segmenting Laser Range Images

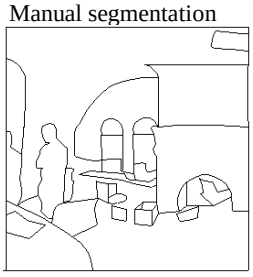
Input range image

Input reflectance image

Our segmentation

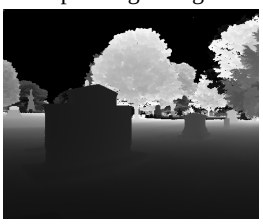


Manual segmentation



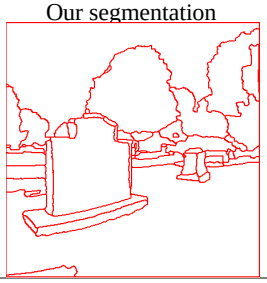
Segmenting Laser Range Images

Input range image

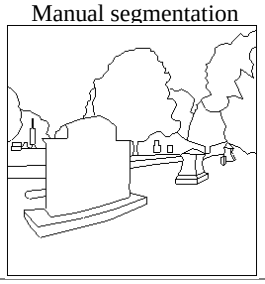


Input reflectance image

Our segmentation



Manual segmentation

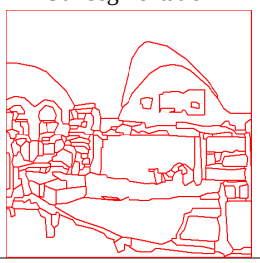


Segmenting Laser Range Images

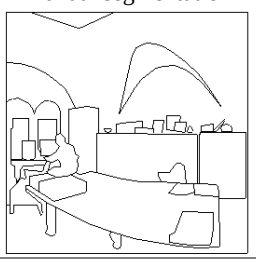
Input range image

Input reflectance image

Our segmentation



Manual segmentation



Two Computing Paradigms in Vision

1. Generative methods --- "Top-down" ← General but quite slow

   explicitly model the visual patterns

   --Bayesian framework,
   --Markov random fields,
   --Markov chain Monte Carlo,
   --Partial differential equations for diffusion, evolving, ...

2. Discriminative methods --- "Bottom-up" ← Fast but not reliable

   explore "intra-class" vs "Inter-class" difference

   -- Feature extraction, on /off, e.g. Edge detection
   -- Data clustering
   -- Adaboost,
   -- Decision tree, …

Summary

1. DDMCMC is a systematic way for integrating "top-down" and "bottom-up".

   The discriminative methods approximate local posterior probabilities (ratios) in various atomic spaces. These probabilities/ratios are used as importance proposal probabilities, and drive the Markov chain to search for globally optimal solutions.

2. Fast Markov chain convergence and mixing at low temperature.

   In contrast to simulated annealing, the SW-type algorithm can move fast at low temperature.

3. Ensemble complexity vs. worst case complexity

   Though one can always construct worst case and prove NP-completeness, but on the average case, the computational complexity can be much lower.

When the bottom-up proposal probabilities fail !