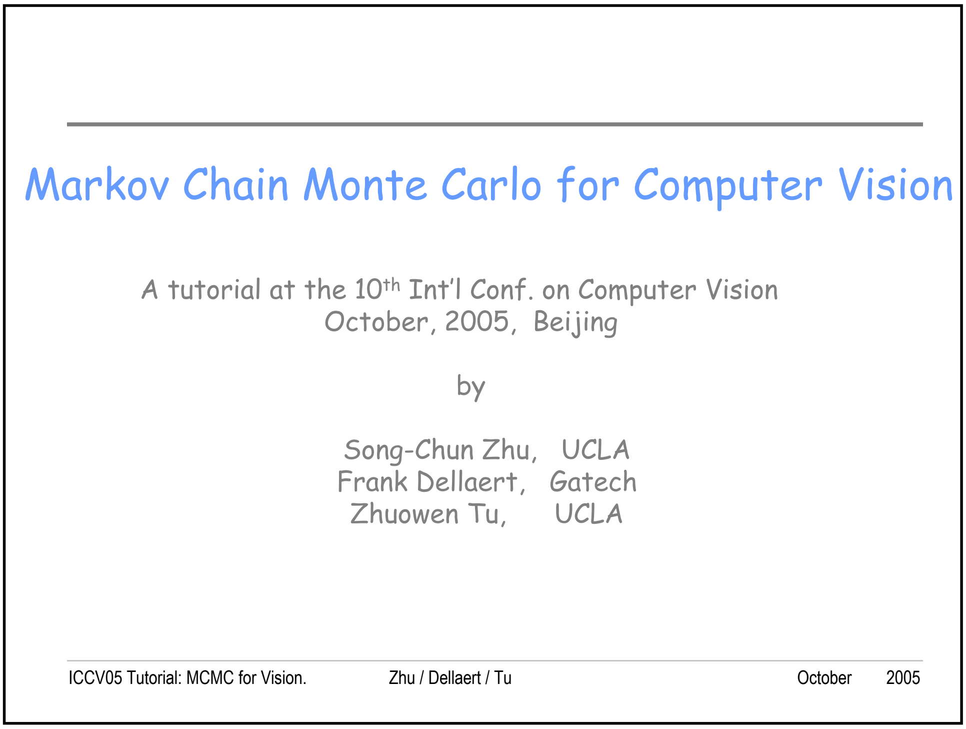
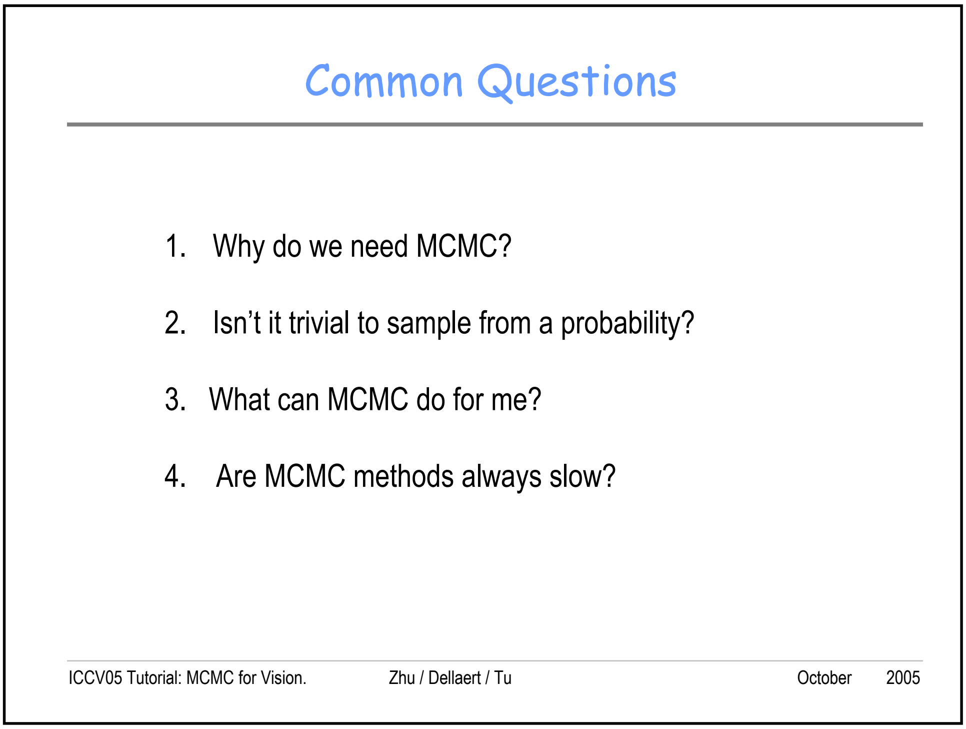


Markov Chain Monte Carlo for Computer Vision

A tutorial at the 10th Int'l Conf. on Computer Vision
October, 2005, Beijing

by

Song-Chun Zhu, UCLA
Frank Dellaert, Gatech
Zhuowen Tu, UCLA



Common Questions

1. Why do we need MCMC?
2. Isn't it trivial to sample from a probability?
3. What can MCMC do for me?
4. Are MCMC methods always slow?

Topics of the tutorial

1. **Introduction to Markov Chain Monte Carlo**
   --- history, concepts, examples, why using MCMC, basics.
2. **Two basic designing tools**
   --- Gibbs sampler, Metropolis-Hastings.
3. **A variety of tricks for MCMC design**
   --- hit-and-run, Metropolized Gibbs, data augmentation, clustering, slice sampling
4. **Reversible jumps**
   --- trans-dimensional sampling, Rao-Blackwellization.
5. **Data-driven Markov Chain Monte Carlo** (DDMCMC)
   ---traversing complex state space, integrating generative and discriminative models.
6. **Cluster sampling**
   --- Swendsen-Wang and generalizations.
7. **Convergence analysis and exact sampling techniques**
   --- convergence rate, exact sampling.

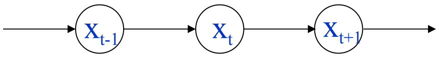
Lect 1: Introduction to MCMC

1. What is Markov chain Monte Carlo?
2. Why using MCMC?
   --- Simulation, optimization, estimation
3. Examples
4. **Computing with two categories of models in vision:**
   --- Descriptive and generative
5. Brief history of MCMC

What is Markov Chain?

A **Markov chain** is a mathematical model for stochastic systems whose states, discrete or continuous, are governed by a transition probability. The current state in a Markov chain only depends on the most recent previous states, e.g. for a 1st order Markov chain.

$$x_t|x_{t-1},\dots,x_0 \ \sim \ P(x_t|x_{t-1},\dots,x_0) = P(x_t|x_{t-1})$$



The **Markovian property** means "locality" in space or time, such as Markov random fields and Markov chain. Indeed, a discrete time Markov chain can be viewed as a special case of the Markov random fields (causal and 1-dimensional).

A **Markov chain** is often denoted by (Ω, ν, K) for state space, initial and transition prob.

What is Monte Carlo ?

Monte Carlo is a small hillside town in Monaco (near Italy) with casino since 1865 like Los Vegas in the US. It was picked by a physicist Fermi (Italian born American) who was among the first using the sampling techniques in his effort building the first man-made nuclear reactors in 1942.

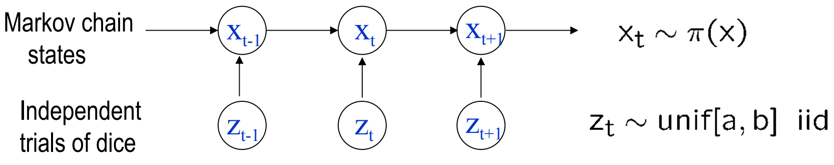
What is in common between a **Markov chain** and the **Monte Carlo casino**?

They are both driven by random variables --- running dice !

Monte Carlo casino

What is Markov Chain Monte Carlo ?

MCMC is a **general purpose technique** for generating **fair samples** from a probability in high-dimensional space, using random numbers (dice) drawn from uniform probability in certain range. A Markov chain is designed to have $\pi(x)$ being its **stationary (or invariant) probability.**



Markov chain states

Independent trials of dice

$x_t \sim \pi(x)$

$z_t \sim \text{unif}[a, b]$ iid

This is a non-trivial task when $\pi(x)$ is very complicated in very high dimensional spaces !

MCMC as a general purpose computing technique

Task 1: Simulation: draw fair (typical) samples from a probability which governs a system.

$$x \sim \pi(x), \text{ s is a configuration.}$$

Task 2: Integration / computing in very high dimensions, i.e. to compute

$$c = E[f(x)] = \int \pi(x) f(x) ds$$

Task 3: Optimization with an annealing scheme

$$x^* = \text{argmax } \pi(x)$$

Task 4: Learning:

unsupervised learning with hidden variables (simulated from posterior)
or MLE learning of parameters $p(x; \theta)$ needs simulations as well.

Task 1: Sampling and simulation

For many systems, their states are governed by some probability models. e.g. in statistical physics, the microscopic states of a system follows a Gibbs model given the macroscopic constraints. The fair samples generated by MCMC will show us what states are ***typical*** of the underlying system. In computer vision, this is often called "***synthesis***" ---the visual appearance of the simulated images, textures, and shapes, and it is a way to ***verify*** the sufficiency of the underlying model.

Suppose a system state x follows some global constraints.

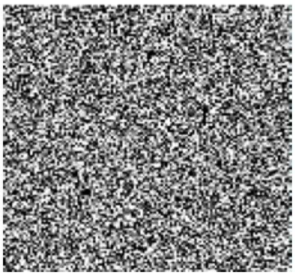
$$x \in \Omega = \{x : H_i(x) = h_i, i = 1, 2, \dots, K\}$$

Hi(s) can be a hard (logic) constraints (e.g. the 8-queen problem), macroscopic properties (e.g. a physical gas system with fixed volume and energy), or statistical observations (e.g the Julesz ensemble for texture).

Ex. 1 Simulating noise image

We define a "noise" pattern as a set of images with fixed mean and variance.

$$\text{noise} = \Omega(\mu, \sigma^2) = \{ I_\Lambda : \lim_{\Lambda \to Z^2} \frac{1}{|\Lambda|} \sum_{(i,j)\in\Lambda} I(i,j) = \mu, \quad \lim_{\Lambda \to Z^2} \frac{1}{|\Lambda|} \sum_{(i,j)\in\Lambda} (I(i,j) - \mu)^2 = \sigma^2 \}$$



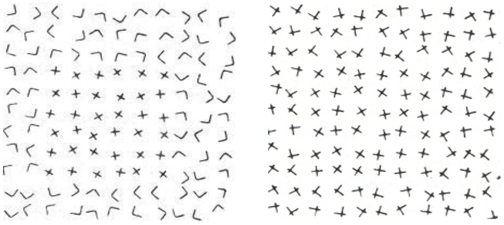
This image example is a "typical image" of the Gaussian model.

Ex. 2 Simulating typical textures

Julesz's quest 1960-80s

"What features and statistics are characteristics of a texture pattern, so that texture pairs that share the same features and statistics cannot be told apart by pre-attentive human visual perception?"

early vision
(0.1-0.4sec)



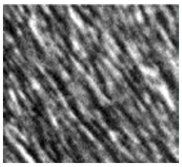
His quest was not answered partly due to the lack of general techniques for generating fair texture pairs that share the same features and statistics, no more no less.

Ex. 2 Simulating typical textures by MCMC

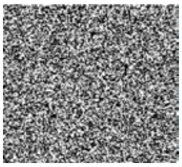
(Zhu et al, 1996-01)

$$\text{a texture} = \Omega(h_c) = \{ I : \lim_{\Lambda \to Z^2} \frac{1}{|\Lambda|} \sum_{(i,j) \in \Lambda} h(I(i,j)) = h_c \,, \quad |h_c| = k \}$$

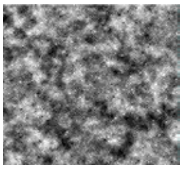
H_c are histograms of Gabor filters, i.e. marginal distributions of $f(I)$



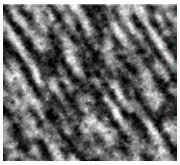
I^{obs}



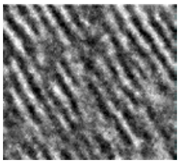
$I^{syn} \sim \Omega(h)$ k=0



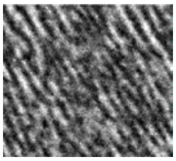
$I^{syn} \sim \Omega(h)$ k=1



$I^{syn} \sim \Omega(h)$ k=3



$I^{syn} \sim \Omega(h)$ k=4



$I^{syn} \sim \Omega(h)$ k=7

Ex 3: Simulating typical protein structures

We are interested in the *typical configurations*, of protein folding given some known properties. The set of typical configurations is often huge !

Molecular dynamcs

Poteintial energy fun$U(x)$

Kinetic ene$K(\dot{x})$

Total energy

$H(x) = U(x) + K(\dot{x})$

Statistical physics

$$x \sim \pi(x) = \frac{1}{Z}\exp\{-\frac{1}{KT}U(x)\}$$

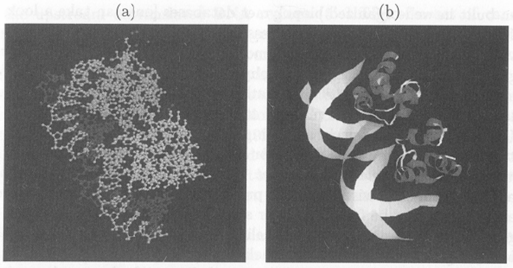


FIGURE 1.4. (a) A ball-and-stick plot of the interaction between a regulatory protein in yeast, 3CRO, and the DNA segment to which it binds. (b) The same structure as in (a), but expressed by a ribbon representation widely used in the protein structure modeling community.

[From ref book by Jun Liu]

Task 2: Scientific computing

In scientific computing, one often needs to compute the integral in very high dimensional space.

Monte Carlo integration,

e.g.

1. estimating the expectation by empirical mean.
2. importance sampling

Approximate counting (so far, not used in computer vision)

e.g.

1. how many non-self-intersecting paths are in a 2 n x n lattice of length N?
2. estimate the value of π by generating uniform samples in a unit square.

Ex 4: Monte Carlo integration

Often we need to estimate an integral in a very high dimensional space Ω,

$$c = \int_{\Omega} \pi(x) f(x) dx$$

We draw N samples from π(x),

$$x_1, x_2, ..., x_N \sim \pi(x)$$

Then we estimate C by the sample mean

$$\hat{c} = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

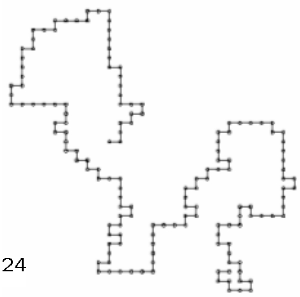
For example, we estimate some statistics for a Julesz ensemble π(x;θ),

$$C(\theta) = \int_{\Omega} \pi(x;\theta) H(x) dx$$

Ex 5: Approximate counting in polymer study

For example, what is the number K of Self-Avoiding-Walks in an n x n lattice?

A Self-Avoiding Walk of Length N=150



Denote the set of SAWs by $\Omega_{n^2} = \{r\}$

An example of n=10. (Persi Diaconis)

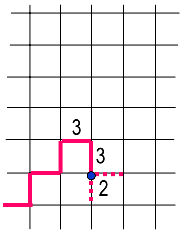
The estimated number by Knuth was $(1.6 \pm 0.3) \times 10^{24}$

The truth number is 1.56875×10^{24}

Ex 5: Approximate counting in polymer study

Computing K by MCMC simulation

$$\begin{aligned} \mathrm{K} &= \sum_{\mathrm{r}\in\Omega_{\mathrm{n}2}} 1 = \sum_{\mathrm{r}\in\Omega_{\mathrm{n}2}} \frac{1}{p(r)} p(r) \\ &= E[\frac{1}{p(r)}] \\ &\approx \frac{1}{M}\sum_{i=1}^{M}\frac{1}{p(r_i)} \end{aligned}$$



Sampling SAWs r_i by random walks (roll over when it fails).

$$\mathrm{p(r)} = \prod_{\mathrm{j}=1}^{\mathrm{m}} \frac{1}{k(j)}$$

Task 3: Optimization and Bayesian inference

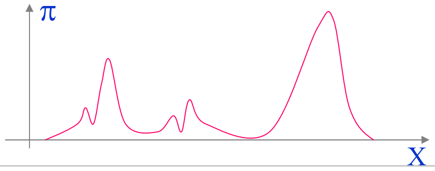
A basic assumption, since Helmholtz (1860), is that biologic and machine vision compute the most probable interpretation(s) from input images.

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

$$\mathrm{X}^* = \arg\max \pi(\mathrm{X}|\mathrm{I})$$

In statistics, we need to sample from the posterior and keep multiple solutions.

$$(\mathrm{X}_1, \mathrm{X}_2, ..., \mathrm{X}_\mathrm{k}) \sim \pi(\mathrm{X}|\mathrm{I})$$

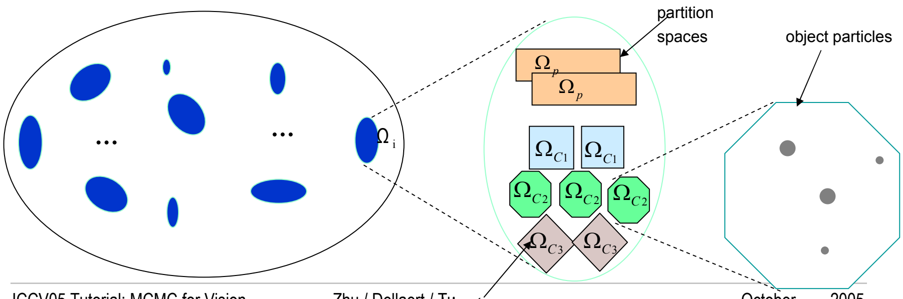


Traversing Complex State Spaces

1. The state space Ω in computer vision often has a large number of sub-spaces of varying dimensions and structures, because of the diverse visual patterns in images.

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

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



Summary

1. MCMC is a general purpose technique for sampling from complex probabilistic models.

2. In high dimensional space, sampling is a key step for
   (a) modeling (simulation, synthesis, verification)
   (b) learning (estimating parameters)
   (c) estimation (Monte Carlo integration, importance sampling)
   (d) optimization (together with simulated annealing).

2. As Bayesian inference have become a major framework in computer vision, the MCMC technique is a useful tool of increasing importance for more and more advanced vision models.

Two categories of graph structures in vision

In computer vision, the target probability $\pi(x)$ is often defined on a graph representation **G=<V, E>.** We divide G in two types of graph structures, and thus the Markov chains are designed accordingly.

1. **Descriptive models** on a plat graph where all vertices are semantically at the same level, e.g. various Markov random fields

   image segmentation, graph partition/coloring, shape from X…

2. **Generative models** on a hierarchic And-Or graph with multiple levels of vertices where a high level vertex is divided into various components at the low level. e.g. Markov trees, sparse coding,

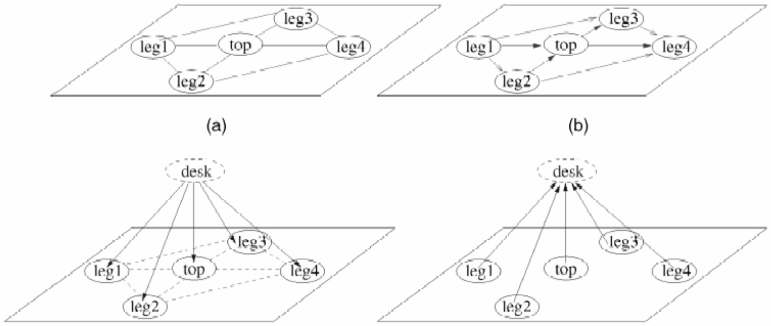
   object recognition, *image parsing, etc*

In advanced models, these two structures are integrated because the vertices at each level of a generative model are connected by contextual horizontal links which Represent various relations among the vertices

To Clarify the terminology

Descriptive or declarative
(Constraint-satisfaction, Markov random fields,
Gibbs, Julesz ensemble)

Variants of Descriptive
(Causal Markov Models,
Markov chain, Markov tree, DAG etc)



Generative (+ Descriptive) (c)
(hidden Markov, hierarchic model
decomposing whole to parts)

Discriminative (d)
(discriminating the whole
using the parts)

MCMC on descriptive models

[Relaxation labeling, belief propagation], Gibbs sampler, Swendsen-Wang, …

Issues in algorithm design:

1. Visiting schedule.

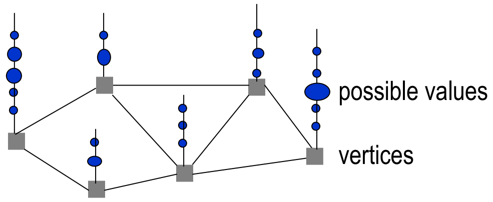
   which vertex is more informative to visit next.

2. Computing joint solution or marginal belief.

   the marginal believe may be conflicting to each other.

3. Clustering strongly-coupled sub-graphs for effective moves.

   the Swendson-Wang ideas.

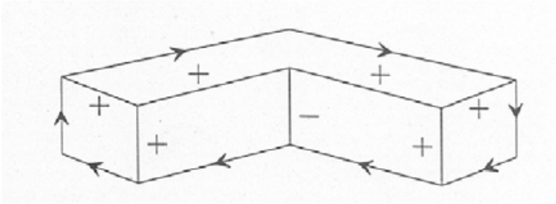


possible values

vertices

Ex 6: Line drawing interpretation

Label the edges of a line drawing (graph) so that they are consistent



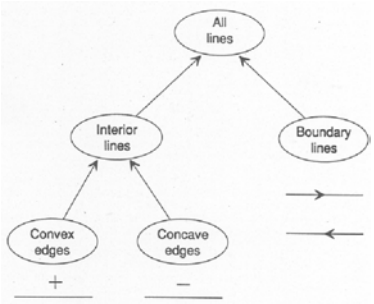
This is also *constraint-satisfaction* problem on a graph G=<V,E>.

$$x \in \Omega = \{x : H_i(x) = h_i, i = 1, 2, ..., K\}$$

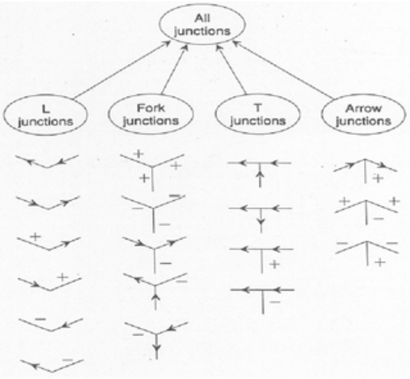
Here each vertex has a set of hard constraints for the labeling of edges ending at the vertex.

Ex 6: Line drawing interpretation

allowed edge labels



allowed junction labels



MCMC on generative models

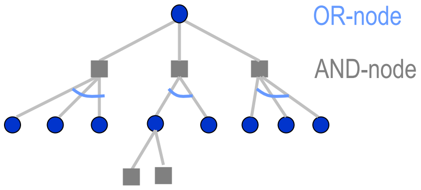
Metropolis-Hastings, Reversible jumps, DDMCMC, top-down / bottom-up message passing

Issues in algorithm design:

1. Reversibility is a concept similar to backtracking in AI search.

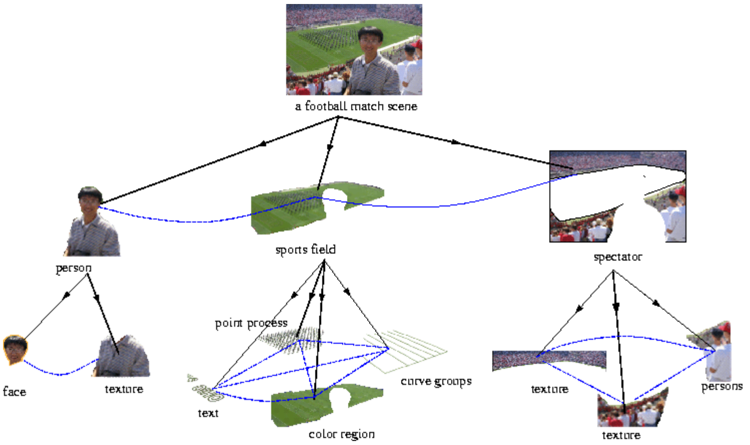
2. Constructing sufficient operators (dynamics) to traverse the entire state space

3. Ordering various Markov chain dynamics (top-down or bottom-up)



Ex 7: Images parsing

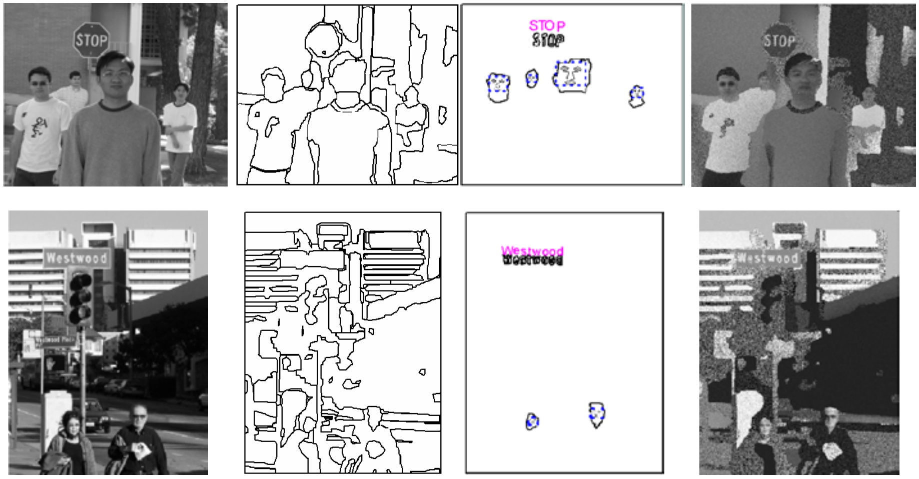
Parsing an image into its constituent visual patterns. The parsing graph below is a solution graph with AND-nodes



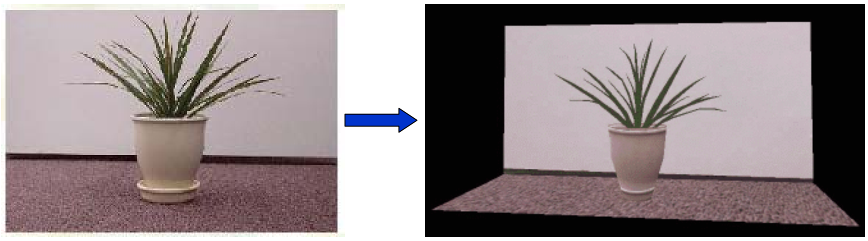
Ex 7: Images parsing

Tu, Chen, Yuille, and Zhu 2003

Input Regions Objects Synthesis



Ex 7. from image parsing to 3D



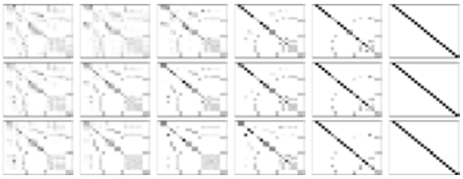
"Bayesian Reconstruction of 3D Shapes and Scenes From A Single Image", F. Han and S.C. Zhu,

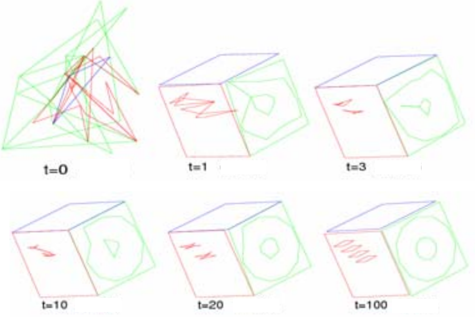
Ex.8 3D Reconstruction via Monte-Carlo EM

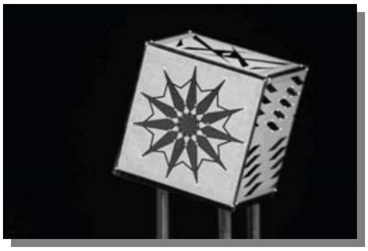
Dellaert, Seitz, Thorpe, & Thrun, 2000

Given: images without correspondence

MCMC inference on very large space of correspondences

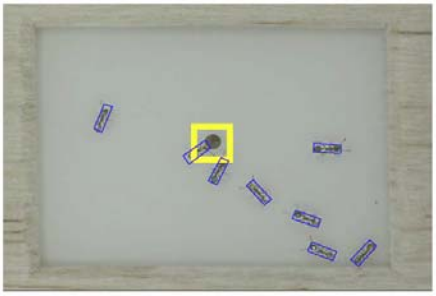






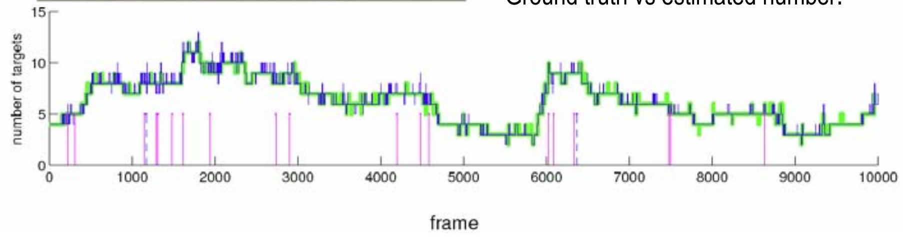
Ex. 9 MCMC-Based Particle Filters

Khan, Balch & Dellaert



Estimating how many targets.

Ground truth vs estimated number.



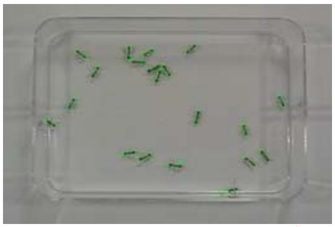
Ex. 9 MCMC-Based Particle Filters

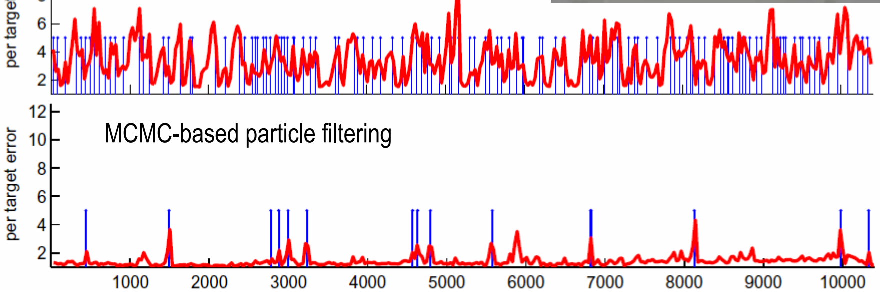
Khan, Balch & Dellaert

Running particle filters in large state spaces (ants, bees, people, sports)

Blue: lose track occurs,

Red: pixel errors per target

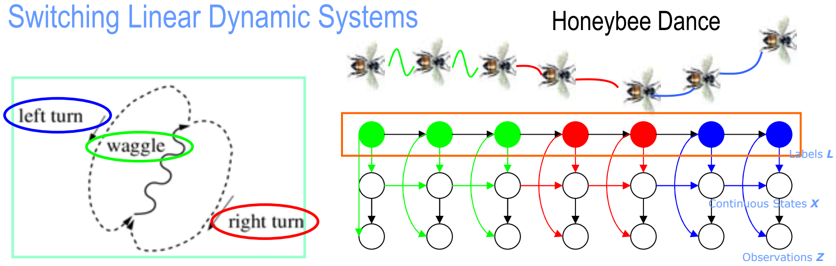


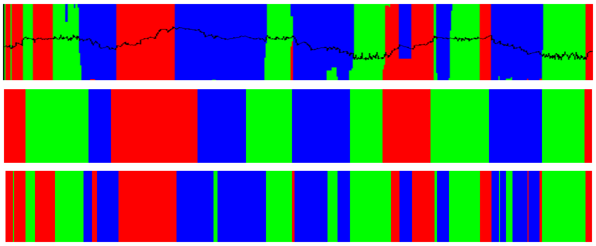


Ex. 10 Inference in SLDS Models

Oh, Rehg, Balch, & Dellaert, AAAI 2005

Switching Linear Dynamic Systems





MCMC MAP label

Ground Truth

Approx. Viterbi

A Brief History of MCMC

1942-46: Real use of MC started during the WWII
--- study of atomic bomb (neutron diffusion in fissile material)

1948: Fermi, Metropolis, Ulam obtained MC estimates for the eigenvalues of the Schrodinger equations.

1950s: Formating of the basic construction of MCMC, e.g. the Metropolis method
--- applications to statistical physics model, such as Ising model

1960-80: Using MCMC to study phase transition; material growth/defect, macro molecules (polymers), etc.

1980s: Gibbs samplers, Simulated annealing, data augmentation, Swendsen-Wang, etc
global optimization; image and speech; quantum field theory,

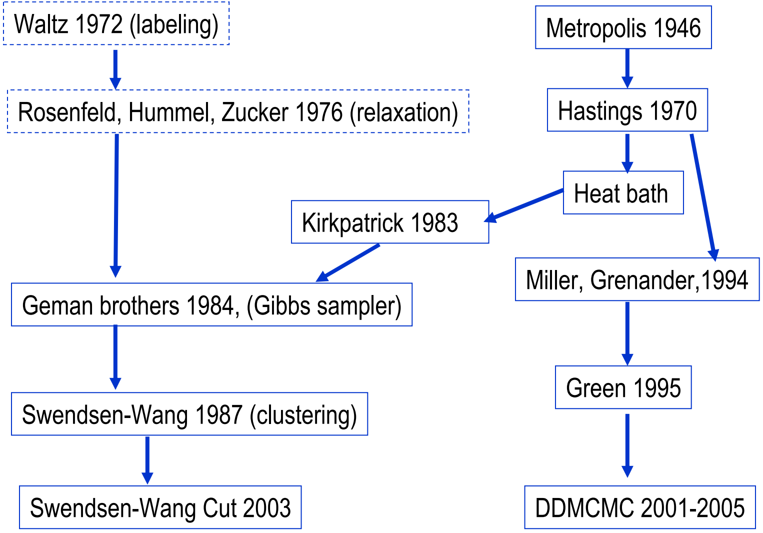
1990s: Applications in genetics; computational biology.

Special cases

When the underlying graph G is a chain structure, then things are much simpler and many algorithms become equivalent.

Dynamic programming (Bellman 1957)
= Gibbs sampler (Geman and Geman 1984)
= Belief propagation (Pearl, 1985)
= exact sampling
= Viterbi (HMM 1967)

Some MCMC developments related to vision


Waltz 1972 (labeling)
Rosenfeld, Hummel, Zucker 1976 (relaxation)
Metropolis 1946
Hastings 1970
Heat bath
Kirkpatrick 1983
Miller, Grenander,1994
Geman brothers 1984, (Gibbs sampler)
Green 1995
Swendsen-Wang 1987 (clustering)
Swendsen-Wang Cut 2003
DDMCMC 2001-2005