MCMC Tutorial at ICCV

8.30-9.15	SCZ	Intro
10.15-11.15	FD	MCMC Basics
10.15-10.45	Break	
10.45-11.30	SCZ	A variety of tricks for MCMC design
11:30-12:15	ZT	Exact sampling techniques
12.15-2.00	Lunch	
2:00-2.45	FD	Trans-dimensional MCMC
2.45-3.30	SCZ	Cluster sampling
3.30-4.00	Break	
4.00-4.45	ZT	Data-driven MCMC

Let5: **Data-driven MCMC**:

Integrating MCMC Search with Discriminative Computing

1. Generative and discriminative models
2. DDMCMC case study
3. DDMCMC approaches for vision applications
4. Conclusions

Challenges

1. Modeling: How to model various patterns in images?

$p(I|W),\ \ likelihood$ — Appearances of scenes are highly complex.

$p(W),\ \ prior\ about\ W$ — Complexity of scene configurations is enormous.

2. Computing: How to make inference of these patterns?

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

Motivation for MCMC

1. Analytical solutions, if available, are always preferred.

Rao-Blackwellization theorem.

2. Otherwise, we should seek to sample p(W|I) directly, if we can.

3. Use proposals to guide the search of Markov chain.

Importance sampling, Metropolis-Hastings,
Slice sampling, Reject sampling,…

Why do we use Metropolis-Hastings?

As an optimization technique: $W^* = argmax\ p(W),\ W \in \Omega$

1. *W* has complicated form and lies in a complex space, Ω, which is composed of sub-spaces of different dimensions.
2. There are no closed form solutions.
3. p(W|I) is not convex and PDE approaches find local minimum.
4. Well-computed proposals can quickly guide the MC jumping among promising modes.

What is the goal of vision? (Freeman and Blake ICCV 2001 short course)

If you are asking,
"Are there any faces in this image?",
then you would probably want to use discriminative methods.

If you are asking,
"Find a 3-d model that describes the runner",
then you would use generative methods.

Generative vs. Discriminative

Bayesian: (Top-down)

$p(W)$ $\quad W = (w_1, w_2, ..., w_k)$

Generation $p(\mathbf{I}|W)$ $\quad$ Inference $p(W|\mathbf{I})$ $\quad$ MCMC sampling

$\mathbf{I}$

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

Data-driven: (Bottom-up)

$W \leftarrow ? \leftarrow (w_1, w_2, ..., w_k)$ (edge, color, face)

$F_1(\mathbf{I})$, $F_2(\mathbf{I})$, $F_k(\mathbf{I})$

$\mathbf{I}$

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

Generative (descriptive) Models: MiniMax Entropy Principle

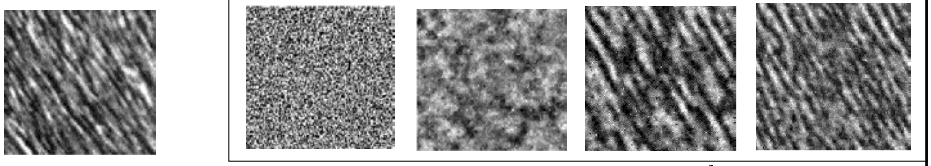
$W = (y, \alpha, \Theta)$

label parameter templates

I are samples of generative model: $I \sim p(x|y, \alpha : \Theta)$

Minimax Entropy Principle: (Zhu, Wu, and Mumford 1997)

$$p_\lambda(I|y) = \frac{1}{\sum_I exp\{-\sum_{j=1}^{T} \lambda_j h_j(I)\}} exp\{-\sum_{j=1}^{T} \lambda_j h_j(I)\},$$



Observation $\quad \rightarrow j$

Discriminative Models: Boosting

AdaBoost and Its Variants: (Freund and R. Schapire 1996, Friedman et al. 1998, Lebanon and Lafferty 2003)

$$p_\lambda(y|I) = \frac{1}{\sum_y exp\{-\sum_{j=1}^{T} \lambda_j f_j(I,y)\}} exp\{-\sum_{j=1}^{T} \lambda_j f_j(I,y)\}$$

$f_{j=1..M}$ are weak classifiers.

MiniMax Entropy Principle and Boosting

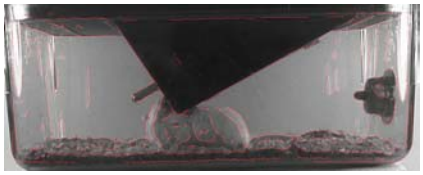
MiniMax Entropy: $$p_\lambda(I|y) = \frac{1}{\sum_I exp\{-\sum_{j=1}^{T} \lambda_j h_j(I)\}} exp\{-\sum_{j=1}^{T} \lambda_j h_j(I)\}$$

Boosting: $$p_\lambda(y|I) = \frac{1}{\sum_y exp\{\sum_{j=1}^{T} \lambda_j f_j(I,y)\}} exp\{\sum_{j=1}^{T} \lambda_j f_j(I,y)\}$$

- ❖ Both have the feature selection procedure (greedy).
- ❖ Both follow the maximum-likelihood principle.

- ▪Generative models focus on single class of interest.
- ▪But Boosting is much easier to use since its normalization term is on y.
- ▪Although generative model is always preferred, we are forced to use discriminative models in many cases.

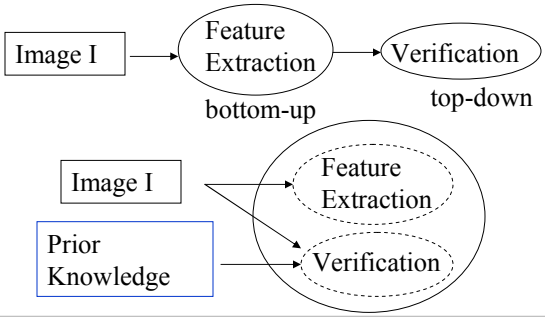
Discriminative models are often not good enough.



Discriminative Models (bottom-up) and Generative Models (top-down)

Methods of integrating between top-down (generative models) and bottom-up information have been widely applied in vision.

Literature: Ullman 1984, Grenander 1993, Belongie et al. 1999, Lee and Mumford 2003, Fei-fei et al. 2003, Fergus et al. 2003, Mori et al. 2004, Yu and Shi 2000, Cremers et al. 2003, Murphy et al. 2004,…



Integrating Discriminative and Generative Models

1. To integrate generative and discriminative models in a principled way.
2. Use discriminative models for fast computation.
3. Use generative models as verification.
4. DDMCMC provides such a framework.

Metropolis-Hastings

To design transition kernel K : $p \bullet K = p$

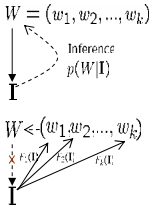
Detailed balance: $p(W_A)K(W_A \to W_B) = p(W_B)K(W_B \to W_A)$

Metropolis-Hastings:

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

$$\alpha(W_A \to W_B) = min\Big(1, \underbrace{\frac{Q(W_A|W_B; F(\mathbf{I}))}{Q(W_B|W_A; F(\mathbf{I}))}}_{hypothesis} \underbrace{\frac{p(W_B|\mathbf{I})}{p(W_A|\mathbf{I})}}_{verification}\Big)$$

$$\underbrace{Q(W_B|W_A)}_{blind\ proposal} \Rightarrow \underbrace{Q(W_B|W_A; F(I))}_{data-driven\ proposal}$$



DDMCMC Approaches for Vision Applications

Target and object recognition:

Grenander and Miller 1994, Zhu et al. 2000, Liu and Shum 2002

Segmentation and Perceptual grouping:

Clark and Quinn 1999, Tu et al. 2001, Ren and Malik 2003, Tu et al. 2003, Barbu and Zhu 2003, Lee and Cohen 2004, Wang et al. 2005, Tu 2005

Tracking:

Tao and Nevatia 2003, Tao and Nevatia 2004, Barbu and Zhu 2004, Oh et al. 2005, . Rittscher et al. 2005

Stereo and 3D:

Dellaert et al. 2001, Han and Zhu 2003, Barbu and Zhu 2005

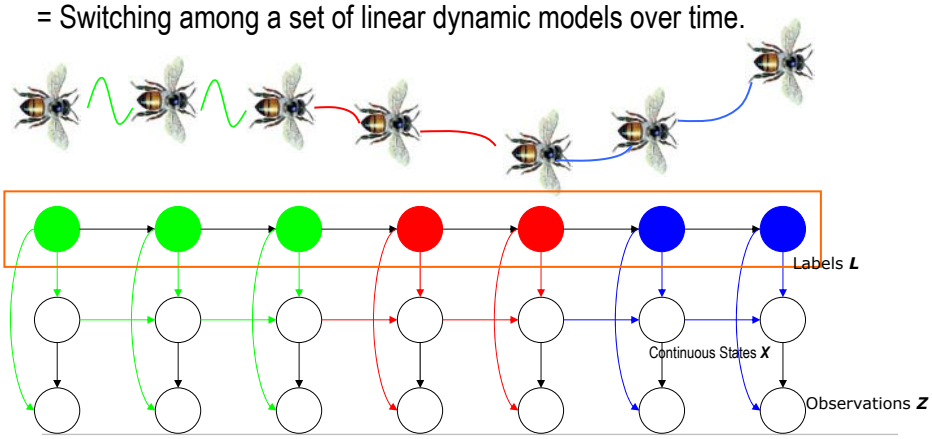
Color constancy: Forsyth 1999

A Case Study: Switching Linear Dynamic Systems

Sang Min Oh, James M. Rehg, Tucker Balch, Frank Dellaert

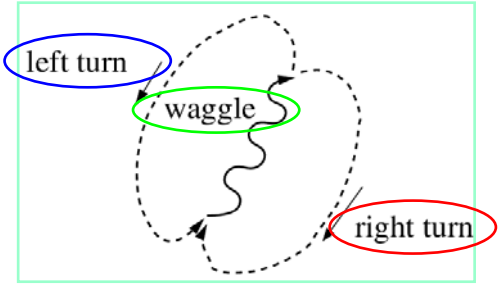
* SLDSs have been widely researched – computer vision *etc.*
* Complex phenomenon

= Switching among a set of linear dynamic models over time.



Honeybee Dance

* Honeybee dance comprises of three patterns.
  : left-turn (blue), waggle (green), right-turn (red)



Inference in SLDS

Inference in an SLDS model is *intractable.* (Lerner & Parr, UAI-01)

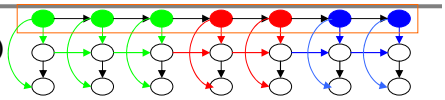
Approximate Inference Algorithms

* Approximate Viterbi, Variational method (Pavlovic & Rehg, CVPR-00)
* GPB2 (Bar-Shalom & Li, 1993)
* Kalman Filtering (Bregler, CVPR-97)
* Expectation propagation (Zoester & Heskes, PAMI-03)
  and many others…

The Ultimate : MCMC inference method.
- Golden standard. Theoretically, it converges to the true posterior.
- Characterizes the accuracy of deterministic Approx. algorithms.

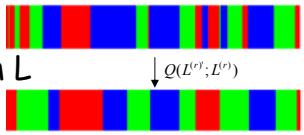
A DDMCMC Approach

* Target Distribution : $P(L \mid Z)$



(1) Start with a valid initial label sequence $L^{(1)}$.

(2) Propose a new label sequence L' from L using a *proposal density* Q .



(3) Calculate the *acceptance ratio* :

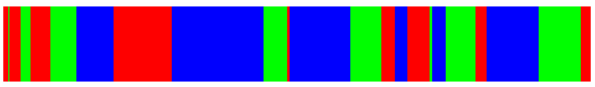
$$a = \frac{P(L' \mid Z)}{P(L \mid Z)} \frac{Q(L; L')}{Q(L'; L)}$$

Rao-Blackwellisation

(4) Accept or reject the new sample in the MH framework.

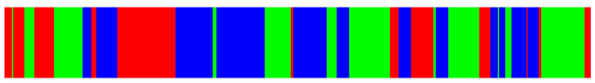
Experimental Results

* Comparison with the approximate Viterbi method.



MCMC MAP label

Ground Truth

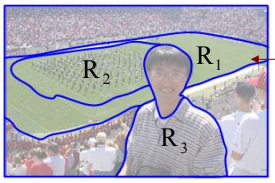


Approx. Viterbi

* Approx. Viterbi VS. MCMC MAP

- A large number of over-segmentations disappear.
- Additionally able to analyze the classification results.

A Case Study: Image Segmentation by DDMCMC

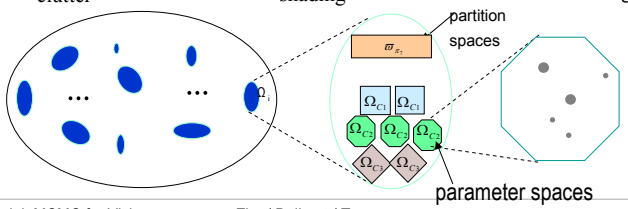


$$W = (n, \{(R_i, l_i, \theta_i), i = 1, ..., n\})$$

R_i: partition (by boundaries)

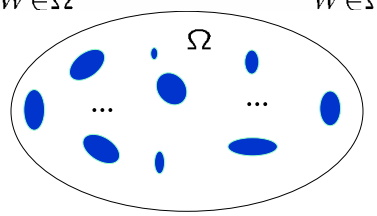
l_i: intensity type $\quad \theta_i$: parameters

grey-scale | color



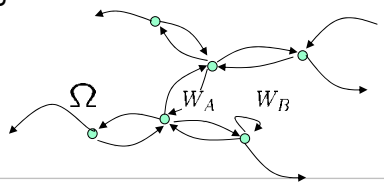
Sampling the Posterior Distribution

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



Markov Chain: $MC = (\pi, K, p_0)$

To design transition kernel: $\pi \bullet K = \pi \quad p_0 \bullet K^n \rightarrow \pi$



Likelihood Models

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, …)

Top-down Approaches

- Gibbs Sampler (Geman and Geman 1984)
- Variational Method (Mumford and Shah 1989)
- Jump-Diffusion (Grenander and Miller 1994)
- Region Competition, PDE (Zhu and Yuille 1996, Osher and Sethian, 1988), …

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

Generation $p(\mathbf{I}|W)$

Inference $p(W|\mathbf{I})$

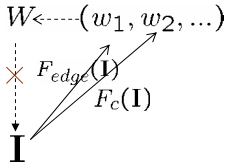
$\mathbf{I}$

Pros: • Easy to incorporate various levels of knowledge.

Cons: • Either local minimal or very slow.

Bottom-up Approaches

- Edge detection (Canny 1986)
- Clustering (Jain and Dubes 1988, Comaniciu and Meer 1998),
- Graph-Cuts (Shi and Malik 1997),…



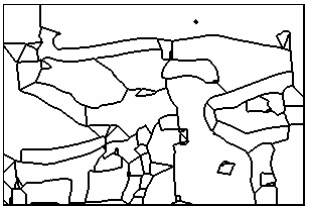
Pros: • Usually very fast.

Cons: • Cues obtained are often local and inconsistent.

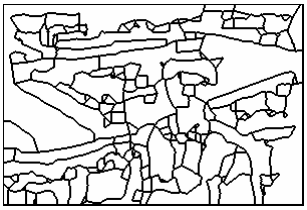
• Data-driven approaches are not directly exploring the solution space.

Proposals by Edge Detection at Different Scales

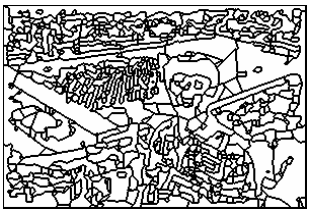
Partition maps: $q(partition|F_{edge}(\mathbf{I}))$



Scale 1



Scale 2



Scale 3

Proposals for 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)

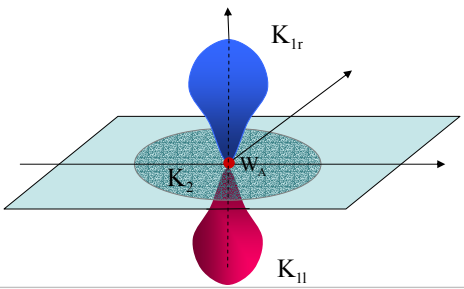
texture

MCMC Kernels

Transition kernel K:

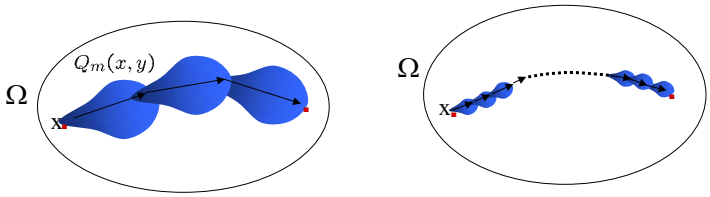
$$K(W_A \to W_B) = \sum_a q_a K_a(W_A \to W_B)$$

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



Design Issues for the Markov Chain Search

1. We want to have efficient moves—big scope of x. $\Omega(x) \sim \Omega$



2. The proposals should be as close to the true distribution as possible. (Generalized Metropolis-Gibbs sampler)

Proposals:

$$Q_m(x,y) = \frac{p(y)/p(x)}{\sum_{y' \in \Omega_m(x)} p(y')/p(x)} \rightarrow \frac{p(y')}{p(x)} \sim q(y'|x)$$

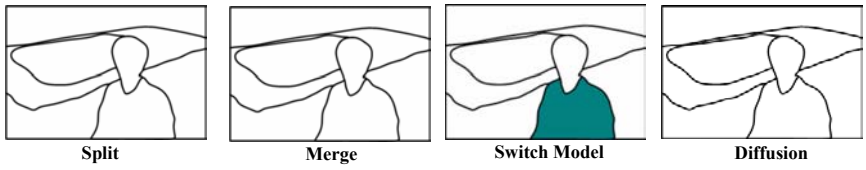
MCMC Moves (Jumps and Diffusions)

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).



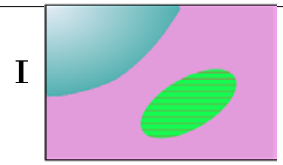
It integrates a variety of existing segmentation methods in computer vision such as:
Edge Detection, Clustering, Split-merge, Region Competition (Snake, MDL, PDE)…

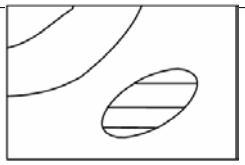
Split and Merge

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

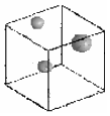
$$\underbrace{K(W_A \to W_B)}_{transition\ probability} = \underbrace{Q(W_B|W_A; F(\mathbf{I}))}_{proposal} \underbrace{\alpha(W_A \to W_B)}_{acceptance\ rate}$$

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

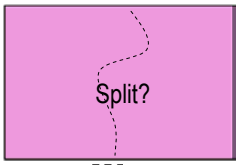




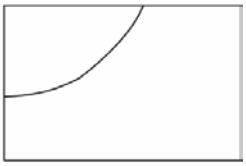
$q(partition|F_{edge}(I))$



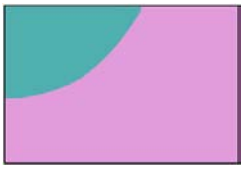
$q(\theta|F_c(I))$



W_A

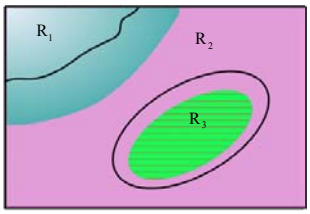


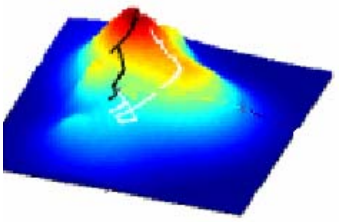
$Q(R_2, R_3|W_A; F_{edge}(I))$



W_B

Stochastic Diffusion and PDE





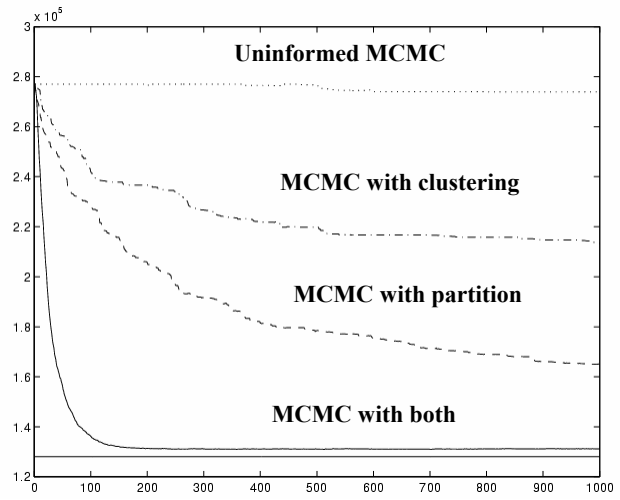
The continuous Langevin equation simulates a Markov Chain with stationary density

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

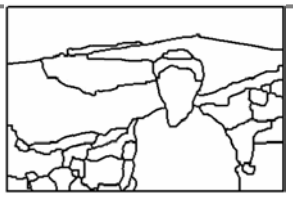
For example, the movement of changing point is driven by

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

Speed Comparison

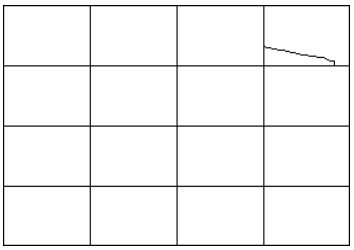


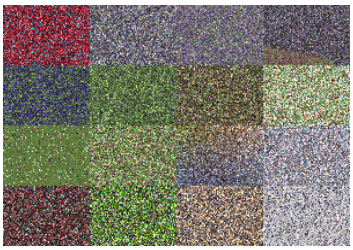
A Demo



Segmentation

Synthesis



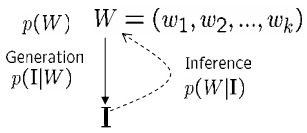


snapshot of solution W sampled by DDMCMC

Revisit of Top-down And Bottom-up

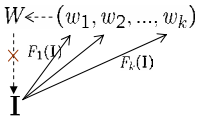
Two approaches: top-down and bottom-up

Top-down (Generative) $p(W|I) \propto p(I|W)p(W)$



- Bayes Theory (Bayes 1763, Pearl 1988, Gelman, Carlin, Stern, and Rubin 1995,...)
- General Pattern Theory (Grenada 1993, Mumford 2001,...)
- ICA, Sparse Coding, Factor Analysis (Common 1994, Olshausen and Filed 1996, Mallat 1989, Hinton 2002,...)

Bottom-up (Discriminative) $q(w_j|F_j(\mathbf{I})) \rightarrow p(w_j|\mathbf{I})$



- Neural Networks (Rosenblatt 1962, LeCun 1986, Rumelhart and McClelland 1986,...)
- Support Vector Machines (Vapnik 1995, ...)
- Boosting, AdaBoost, Bagging (Freund and Schapire 1996, Friedman et al.1998,...)
- Decision Tree (Wang and Suen 1984, Amit and Geman 1997,...)

Some Failure Examples: Need to Engage Middle-level and High-level Knowledge

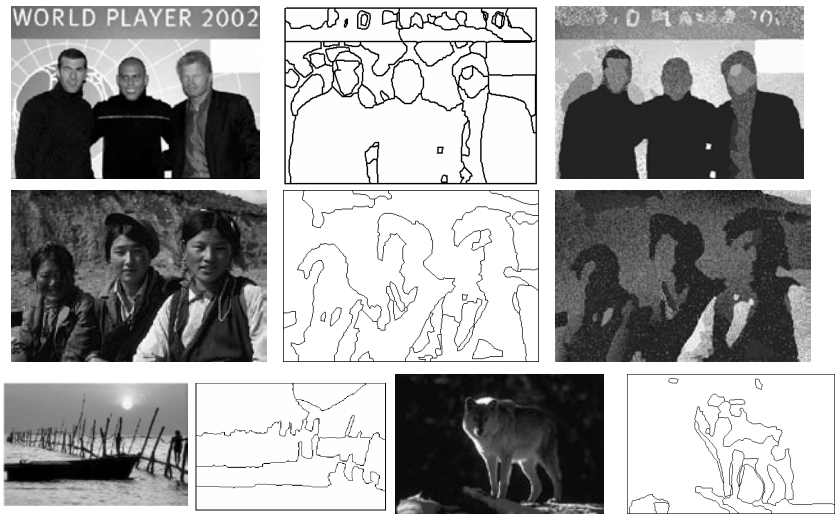


Image Parsing

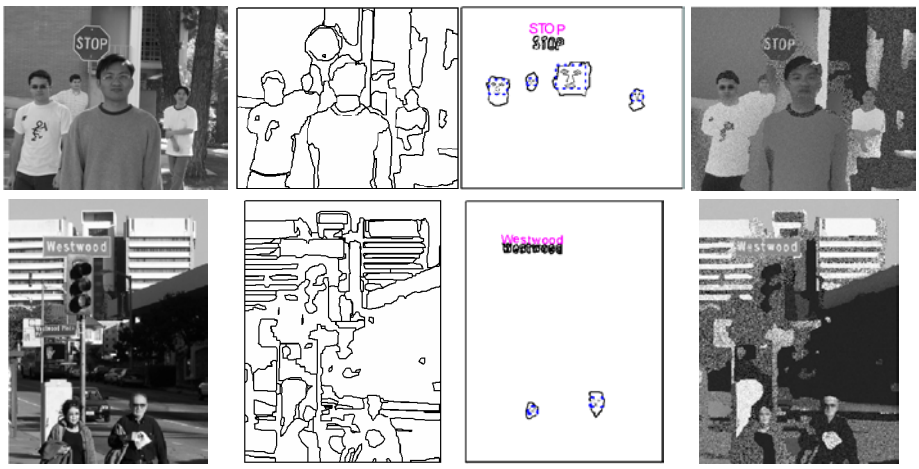
$$W = (n, \{(\zeta_i, R_i, l_i, \theta_i), i = 1, ...n\})$$

Face images of FERET dataset

Text images of San Francisco street scenes.

Results

Input Regions Objects Synthesis

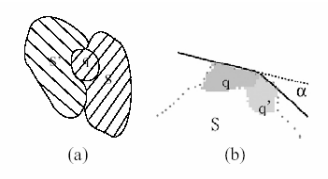


Applications—Segmentation with Gestalt Laws

X. Ren and J. Malik, **Learning a Classification Model for Segmentation,** ICCV 2003.

To integrate the low level segmentations with mid-level Gestalt cues.

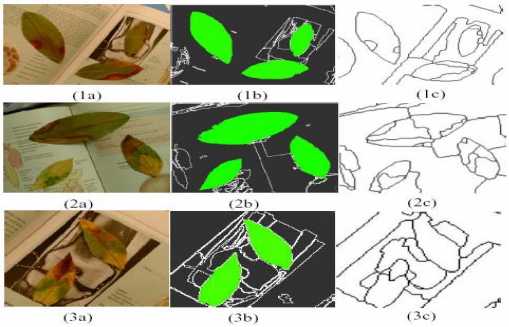
$$f(S) = \sum_{S \in S} (\sum_{j} c_j F_j(S) - \theta)$$



Applications—SWC with Shape Prior

J. Wang, E. Gu, and M. Betke,"**MosaicShape: Stochastic Region Grouping with Shape Prior**", CVPR 2005.

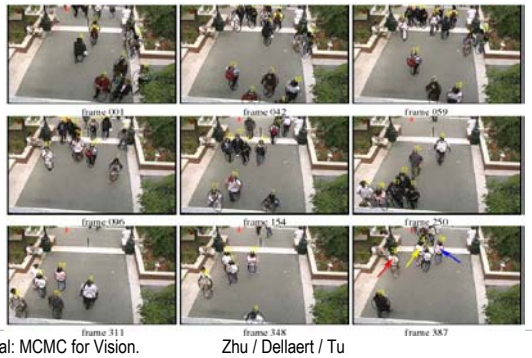
$$\begin{aligned} p(W|I,S) &\propto p(I|S,W)p(S|W)p(W) \\ &\propto [\prod p(I_{V_i}|V_i,S)][\prod p(S|V_i)]\, p(W) \\ &\propto [\prod p(I_{V_i}|\theta_i,S)][\prod p(S|C_{V_i})]\, p(W) \end{aligned}$$



Applications—Tracking Multiple Objects

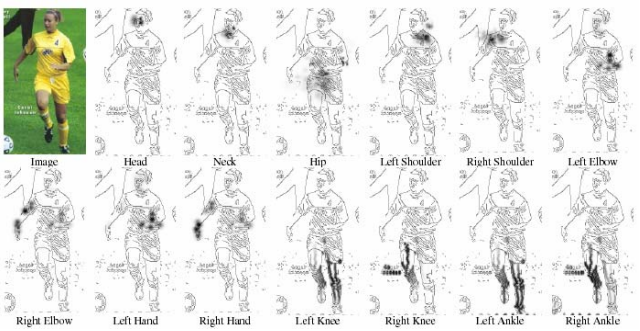
T. Zhao and R. Nevita, "**Tracking Multiple Humans in Crowded Environment**", CVPR 2004.

$$\theta^{(t)*} = argmax\ p(\theta^{(t)}|I^{(t)},\theta^{(t-1)},Bg^{(t-1)})$$
$$= argmax\ p(I^{(t)}|\theta^{(t)},Bg^{(t-1)})p(\theta^{(t)}|\theta^{(t-1)})$$



Applications—Body Configuration

M.W. Lee and I. Cohen, "**Proposal Maps driven MCMC for Estimating Human Body Pose**", CVPR 2004.



proposals

results

Theoretical Side

Analyze the convergence rate of the system.

Theorem

Metropolized Independence Sampler:

$$\frac{1}{min\{p(x), q(x)\}} \leq E[\tau(x)] \leq \frac{1}{min\{p(x), q(x)\}} \cdot \frac{1}{1 - ||p - q||}$$

Maciuca and Zhu 2003

To study the optimal control strategy.

- Ordering of the kernels given the current cues (tests).

$$W_t \sim \mu_t(W) = \nu(W_0) \circ K_{a(1)} \circ K_{a(2)} \circ \cdots \circ K_{a(t)}$$
$$\delta_{a(t)} \stackrel{def}{=} KL(p(W|\mathbf{I})||\mu_t(W)) - KL(p(W|\mathbf{I})||\mu_{t+1}(W)) = E[KL(K_{a(t)}(W_t|W_{t+1})||p_{MC}(W_t|W_{t+1}))]$$

- Ordering the tests by their power.

$$\begin{aligned}\delta(w|F_+) &= E[KL(p(w|\mathbf{I})||q(w|Tst_t(\mathbf{I})))] - KL(p(w|\mathbf{I})||q(w|Tst_t(\mathbf{I}), F_+)) \\ &= MI(w||Tst_t(\mathbf{I}, F_+) - MI(w||Tst_t(\mathbf{I})) = KL(q(w|Tst_t(\mathbf{I}), F_+)||q(w|Tst_t(\mathbf{I})))\end{aligned}$$

- Ordering kernels and tests by their computation cost.

Take-home Messages for DDMCMC Approaches

1. **DDMCMC is an open framework which can integrate various of methods (top-down, bottom-up, and PDEs).**
2. **DDMCMC can deal with solutions of complex form.**
3. **The performance of DDMCMC is largely decided by proposals.**
4. **To decrease the complexity of the approach and increase its scalability.**
5. **We should tightly couple discriminative and generative models and bring more learning aspects to DDMCM.**