Proceedings of Int'l Conf. on Computer Vision and Pattern Recognition, South Carolina, June, 2000.

Integrating Bottom-Up/Top-Down for Object Recognition by Data Driven Markov Chain Monte Carlo

Song-Chun Zhu, Rong Zhang, and Zhuowen Tu

Department of Computer and Information Sciences
The Ohio State University, Columbus, OH 43210
$\{szhu, rong, ztu\}$@cis.ohio-state.edu

Abstract

This article presents a mathematical paradigm called Data Driven Markov Chain Monte Carlo (DDMCMC) for object recognition. The objectives of this paradigm are two-fold. Firstly, it realizes traditional "hypothesis-and-test" methods through well-balanced Markov chain monte Carlo (MCMC) dynamics, thus it achieves robust and globally optimal *solutions. Secondly, it utilizes data-driven (bottom-up) methods in computer vision, such as Hough transform and data clustering, to design effective transition probabilities for Markov chain dynamics. This drastically improves the effectiveness of traditional MCMC algorithms in terms of two standard metrics: "burn-in" period and "mixing" rate. The article proceeds in three steps. Firstly, we analyze the structures of the solution space Ω for object recognition. Ω is decomposed into a large number of subspaces of* varying dimensions *in a hierarchy. Secondly, we use data-driven techniques to compute* importance proposal probabilities *in these spaces, each expressed in a non-parametric form using weighted samples or particles. Thirdly, Markov chains are designed to travel in such heterogeneous structured solution space, with both jump and diffusion dynamics. We use possibly the simplest objects - the "Ψ-world" as an example to illustrate the concepts, and we briefly present results on an application of traffic sign detection.*

1 Introduction

Over the past 40 years, Markov chain Monte Carlo (MCMC) method has penetrated many subjects, such as statistical physics, seismology, chemistry, biometrics, protein folding, as a general engine for inference and optimization. As a matter of fact, it is the only known method for global optimization, beating genetic algorithm. However, in computer vision the promise of MCMC method has been shadowed by two shortcomings despite many early work[4, 6]. Firstly, it lacks a rigorous mathematical framework for designing well-balanced MCMCs when the solution spaces become complicated and heterogeneously structured. Secondly, naive MCMC algorithms borrowed from statistics and physics are extremely ineffective – reflected by unacceptably long "*burn-in*" periods and incredibly slow "*mixing*" rates.

In this article, we argue that the inefficiency of current MCMC algorithms is not a result of any intrinsic limits of the MCMC methods , but a result of lacking domain knowledge in designing MCMC.

We propose a new mathematical paradigm called *Data Driven Markov Chain Monte Carlo* (DDMCMC). DDMCMC utilizes bottom-up (data driven) techniques, such as Hough transform and data clustering, to compute coarse probabilities which guide the Markov chain to search in promising places or hot spots in solution spaces. The DDMCMC paradigm achieves two objectives: Firstly, it realizes traditional engineering methods of "hypothesis-and-test" through well-balanced Markov chain dynamics, therefore obtains *robust and globally optimal* solutions. Secondly, it drastically improve the effectiveness of traditional MCMC algorithms.

This paper presents a pilot project for object recognition as a proof of concept in a sequence of papers[11, 12]. We proceed in three steps. Firstly, we analyze explicitly the structures of the solution space Ω for object recognition. This leads to the decomposition of Ω–which we call the *scene space* into a large number of subspaces of *varying dimensions*. These subspaces are, in turn, decomposed as *object spaces*, and the latter are production spaces of a number of *atomic spaces*. The atomic spaces are for basic visual cues and are not further decomposable. Secondly, we adopt data-driven techniques in computer vision to compute *importance proposal probabilities* in the atomic and object spaces. These probabilities are expressed in non-parametric forms by a set of weighted samples or *particles*. These particles are respectively called *atomic particles* and *object particles*. Thirdly, Markov chains are designed to travel in such heterogeneously structured solution space, with both jump and diffusion dynamics. These dynamics in combination form reversible, ergodic, and aperiodic random walks that sample the posterior probability in the Bayes framework. We shall also discuss how we compute *multiple, distinct solutions*– which we call *scene particles* in Ω. A key observation is that in this DDMCMC paradigm, we dynamically maintain a finite set of particles at all levels in the hierarchy of the solution space.

Our work is inspired by a serial of recent success in using MCMC methods, including Condensation algorithm for object tracking[7], the stochastic texture modeling and medial axis[10, 9] and others[8].

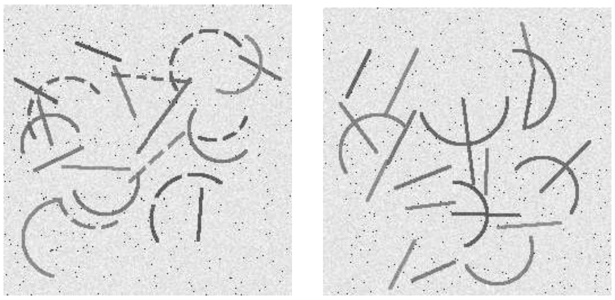
2 The structures of solution space

we use possibly the simplest example – the "Ψ-world" to illustrate the concepts of DDMCMC. The "Ψ-world" consists of only four types of objects: background pixels, line segments, circular arcs, and the Greek symbol Ψ, labeled by B, L, C, P respectively. The observed images $\mathbf{I}$ on a lattice Λ are generated through superposing n objects on the background with additive Gaussian noise. n is subject to a Poisson distribution, and the sizes and locations of the objects are subject to some uniform distributions. Figure 1 displays two *typical images* under the specified distributions.

The Ψ-world is described by a vector,

$$W = (n, \{(\ell_i, \theta_i); i = 0, 1, ..., n, \;\; \boldsymbol{\alpha}).$$

$n \in \{0, 1, 2, .., |\Lambda|\}$ is the number of objects other than background, with $|\Lambda|$ the number of pixels in an image. $\ell_i \in \{B, L, C, P\}$ is the label and θ_i is a vector valued parameter describing the i-th object. We have only only background object, $\ell_0 = B$.



a b

Figure 1: Some examples of the randomly generated images with varying number of objects.

The parameters are defined in the following.

1. Type B: θ_0 is just μ_0 for the grey level of pixels.

2. Type L: θ_i includes $(\rho_i, \tau_i, s_i, e_i, \mu_i)$. (ρ_i, τ_i) describes a straight line, and s_i, e_i the starting and ending points. μ_i is the intensity level of the line.

3. Type C: θ_i includes $(x_i, y_i, r_i, s_i, e_i, \mu_i)$ denoting the center, radius, starting point, ending point, and intensity level for arc objects.

4. Type P: θ_i includes $(x_i, y_i, r_i, \tau_i, \mu_i)$ denoting the center and radius of the half circle and the angle of the line segment, and the intensity level. By definition, the arc in Ψ must be a half circle.

Another important variable in W is the $\boldsymbol{\alpha}$-map for occlusion.

$$\boldsymbol{\alpha} \; : \; \Lambda \rightarrow \{0, 1, 2, ..., n\}, \quad \boldsymbol{\alpha} \in \Omega_{\boldsymbol{\alpha}}.$$

For a pixel $(x, y) \in \Lambda$, $\boldsymbol{\alpha}(x, y)$ indexes the top-most object which is the only visible object at this pixel.

We denote by $\varpi_g = [0, 255]$ the space for image intensity levels, and the solution space of the Ψ-world is,

$$\Omega = \Omega_{\boldsymbol{\alpha}} \times \varpi_g \times \cup_{n=0}^{|\Lambda|} \Omega_o^n,$$

where Ω_o^n is the subspace with exactly n objects excluding the background.

$$\Omega_o^n = \cup_{k+l+m=n} \Omega_{k,l,m}, \;\; k, l, m \geq 0,$$

where $\Omega_{k,l,m}$ is the subspace that has exactly k lines, l arcs, and m Ψ-objects respectively.

$$\Omega_{k,l,m} = \underbrace{\Omega_L \times \cdots \times \Omega_L}_{k} \times \underbrace{\Omega_C \times \cdots \times \Omega_C}_{l} \times \underbrace{\Omega_\Psi \times \cdots \times \Omega_\Psi}_{m}.$$

We call Ω_L, Ω_C, and Ω_Ψ the *object spaces*.

These object spaces are further decomposed into five *atomic spaces*, which are denoted by lower case Greek symbols.

1. ϖ_g: the space for pixel intensity μ.
2. ϖ_c: the space for circle variables x, y, r.
3. ϖ_l: the space for line variables ρ, τ.
4. ϖ_e: the space for starting and ending points s, e.
5. ϖ_τ: the space for the orientation of Ψ.

Thus the object spaces are products of atomic spaces.

$$\Omega_l = \varpi_l \times \varpi_e \times \varpi_g, \quad \Omega_c = \varpi_c \times \varpi_e \times \varpi_g, \quad \Omega_\Psi = \varpi_c \times \varpi_\tau \times \varpi_g.$$

Figure 2 illustrates the structures of the solution space Ω. The triangles, squares and hexagons represent the three object spaces Ω_L, Ω_C, and Ω_Ψ respectively. The small circles of various shadows and sizes represent the five atomic spaces. The arrows indicate

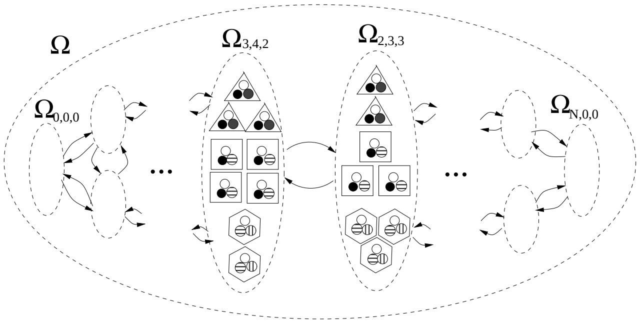


Figure 2: The solution space Ω for the Ψ-world.

the reversible jumps between subspaces discussed in the next section.

Object recognition in the Ψ-world is posed as a Bayes inference problem,

$$W \sim p(W|\mathbf{I}) \propto p(\mathbf{I}|W)p(W), \quad W \in \Omega.$$

$p(W)$ is the product of a Poisson distribution on the object number n and some uniform densities over the object parameters $\theta_i, i = 1, ..., n$. Each W with all the parameters θ_i and $\boldsymbol{\alpha}$-map deterministically specify a clean image $\mathbf{I}_o$, and $p(\mathbf{I}|W) = p(\mathbf{I}|\mathbf{I}_o)$ is simply the product of iid Gaussian noise. Due to space limitation, we choose not to define the probabilities in details.

Notice that *the probability mass of $p(W|\mathbf{I})$ is distributed over many subspaces of varying dimensions*, in the next section, we simulate stochastic Markov chain dynamics which can travel in such heterogeneously structured space and achieve two general goals.

1. To compute the global optimum W^* in Ω.

2. To compute M distinct solutions (or explanations) $S = \{W_1, W_2, ..., W_M\}$ for robustness.

3 Designing MCMC–the basic issues

There are two basic requirements for designing MCMC dynamics.

Firstly, it should be ergodic and aperiodic. Given any two points $W, W' \in \Omega$, the Markov chain can travel from W to W' in finite steps.

Secondly, it should observe the stationary equation. This requirement is often replaced by a stronger condition: the so-called *detailed balance equations*. For any $W, W' \in \Omega$

$$p(W|\mathbf{I})dWP(W \to dW') = p(W'|\mathbf{I})dW'P(W' \to dW). \quad (1)$$

$P(W \to W')$ is the *transition probability*, which governs the Markov chain. Equation (1) requires every move of the Markov chain to be immediately reversible and balanced.

Let $|\Omega|$ denotes the number of discretized solutions, then $p(W|\mathbf{I})$ is a known $|\Omega| \times 1$ vector, and $P(W \to W')$ is an unknown $|\Omega| \times |\Omega|$ matrix that ought to be designed. Equations (1) specify $|\Omega|$ constraints. The transition probability should also satisfy $|\Omega|$ normalization conditions, $\sum_{W'} P(W \to W') = 1, \forall W \in \Omega$. Thus we have $2|\Omega|$ constraints and $|\Omega|^2$ unknown variables in $P(W \to W')$. *In general, there are countless designs for transition probabilities $P(W \to W')$ or Markov chains.*

In vision literature, naive MCMC algorithms borrowed from statistics and physics are bound for inefficiency because they were NOT originally designed for vision problems. Intuitively, in a complicated solution space, an arbitrarily designed MCMC is like a drunk man walking in a cluttered building in the dark. A good design of MCMC should illuminate the building by studying the structures of the solution space and by exploiting the domain knowledge coded in $p(W|\mathbf{I})$ – heuristics.

We design MCMC in two steps.

Firstly, to meet the first requirement of MCMC design, we choose five types of MCMC dynamics.

Type I: diffusion process. This process changes the parameters θ_i, e.g. moving and extending a line segment etc.

Type II: death process. This process eliminates an existing object, and jumps to a subspace of lower dimensions.

Type III: birth process. This process adds a new object, and jumps to a subspace of higher dimensions.

Type IV: composition process. This process composes two existing objects into a new object, and jumps to a different subspace. For example, composing two short lines into a long line, or combining a line with a circle into a Ψ.

Type V: decomposition process. This process decomposes an existing object into two.

For example, in figure 2 the move from $\Omega_{3,4,2}$ to $\Omega_{2,3,3}$ composes a line object and a circle object into a Ψ-object.

The five types of dynamics are applied in random sequence decided by throwing dice. It is easy to prove that Markov chain with the five types of moves is reversible, ergodic, and aperiodic.

Secondly, we discuss how to balance the Markov chain dynamics. The diffusion process can be implemented by the stochastic Langevin equation, which is a steep ascent dynamics maximizing $p(W|\mathbf{I})$ with respect to θ_i plus a Brownian motion. It can also be implemented by a continuous Gibbs sampler.

Due to space limitation, we only discuss the balance between types II and III. The dynamics of type IV and V can be done in a similar way. Suppose at a certain time step t, we *propose* to eliminate an existing arc object specified by $\theta_i = (x_i, y_i, r_i, s_i, e_i, \mu_i)$:

$$W = (n, \theta_i, w) \longrightarrow (n-1, w) = W'.$$

w denotes the objects which remain unchanged during this move. In order to realize the death move, we must compute *how likely the same object can be burn immediately – an inverse move by the birth process.* Notice that this is a pair of moves that jump between two subspaces of different dimensions. We use the Metropolis-Hastings method. Let $G(W \to dW')$ and $G(W' \to dW)$ be the proposal probabilities for the two moves respectively, then the death move is accepted with probability,

$$A(W \to dW') = \min(1, \frac{G(W' \to dW)p(W'|\mathbf{I})dW'}{G(W \to dW')p(W|\mathbf{I})dW}). \tag{2}$$

The transition probability $P(W \to dW') = G(W \to dW')A(W \to dW')$ for $W \neq W'$. The ratio of the posterior probabilities is often the dominating factor which erects the influence of the posterior to balance the possible biases from the proposal probabilities.

The death proposal probability is

$$G(W \to dW') = q(\mathrm{II})q_o(i)dw. \tag{3}$$

$q(\mathrm{II}) \in (0,1)$ is the probability for using type II dynamics at time t, and $q_o(i)$ is the probability for choosing the circle object θ_i. The birth proposal is

$$G(W' \to dW) = q(\mathrm{III})q(\theta_i)d\theta_i dw. \tag{4}$$

It first chooses type III with probability $q(\mathrm{III})$, then chooses a new circle object θ_i with probability $q(\theta_i)d\theta_i$.

Since $dW = d\theta_i dw$ and $dW' = dw$, the dimensions of the nominator and denominator in equation (2) are matched. Designing $q(\mathrm{II})$ and $q(\mathrm{III})$ is easy and often not crucial for the speed. For example, one may use type II more frequently at the beginning. The key issue here is to compute $q(\theta_i)$!

In the statistical literature, the jump-dynamics were first studied in [6, 5], where variables in new dimensions are proposed by prior models. In our case, this chooses $q(\theta_i)$ to be a uniform distribution like blind search. Obviously such proposals are most likely to be rejected. This is a main reason for the MCMC to be inefficient! Intuitively, $q(\theta_i)$s should be able to predict where the new objects are likely to be in the object spaces. This is where the data driven (bottom-up) techniques step in.

4 Computing proposal probabilities in the atomic spaces – atomic particles

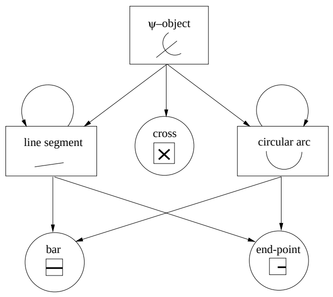


Figure 3: Hierarchy in the Ψ world.

Figure 3 displays the hierarchy of objects in the Ψ-world. The terminal (circles) nodes represent the elements of features: bar, end point, and cross, and arrows mean compositional relations. We use three types of feature detectors displayed in figure 4. There are 3 cross detectors, 6 bar detectors, and 12 end point detectors at various orientations.

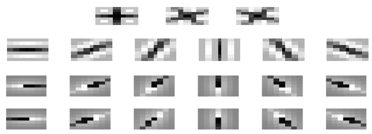
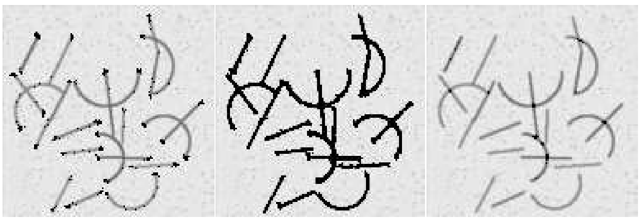


Figure 4: Feature detectors: 3 for cross (row 1), 6 for bar (row 2), and 12 for end point (row 3,4).

Figure 5 displays the results of three feature maps of bars, end points, and crosses for an image in figure 1.b. Dark points in figure 5 have higher responses. At each location, we show the orientation which has maximum response.



a b c

Figure 5: The results of a). bar detection, b). end point detection, c). cross detection.

Using the bar-detection map, we compute the Hough transforms for both lines and circles. Figure 6 displays the Hough transform result for straight lines in the image shown in figure 1.b. The darker a point is, the more votes it accumulates. Local maxima are computed using a mean-shift algorithm[1], and are marked by the crosses in the figure. We denote these local maxima by $(\rho_l^{(i)}, \tau_l^{(i)})$, $i = 1, 2, ..., n_l$.

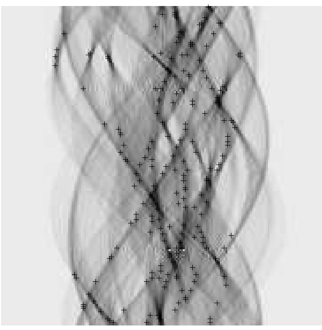


Figure 6: The result of Hough transform for lines (ρ, θ), and sample points: line particles.

Thus we compute an empirical density on the atomic space ϖ_l, expressed as a set of weighted samples.

$$q_l(\rho, \tau) = \sum_{i=1}^{n_l} w_l^{(i)} \delta(\rho - \rho_l^{(i)}, \tau - \tau_l^{(i)}), \sum_{i=1}^{n_l} w_l^{(i)} = 1.$$

$\delta(\rho - \rho_l^{(i)}, \tau - \tau_l^{(i)})$ is a window function centered at $(\rho_l^{(i)}, \tau_l^{(i)})$. We call $q_l(\rho, \tau)$ the *importance proposal probability* in atomic space ϖ_l, and $\{(\rho_l^{(i)}, \tau_l^{(i)}), i = 1, 2, ..., n_l\}$ the *atomic particles.*

Similarly figure 7 displays the atomic particles in space ϖ_c. They are the local maxima of the Hough transform results for circles. The sizes of the spheres represent the weights $w_c^{(i)}$.

So we have an atomic proposal probability on ϖ_c,

$$q_c(x, y, r) = \sum_{i=1}^{n_c} w_c^{(i)} \delta(x - x_c^{(i)}, y - y_c^{(i)}, r - r_c^{(i)}), \; \sum_{i=1}^{n_c} w_l^{(i)} = 1.$$

In a similar way, one can compute proposal probabilities in other atomic spaces. 1) In ϖ_g we compute the intensity histogram $q_g(\mu)$. 2) In ϖ_e, we have computed the end-point maps. 3) In ϖ_τ, we can simply project $q_l()$ onto the τ axis. *For robustness and reversibility, the atomic proposal probabilities are continuous and non-zero everywhere in atomic spaces.*

5 Computing proposal probabilities in object spaces– object particles

Because the object spaces are products of atomic spaces, proposal probabilities in the three object

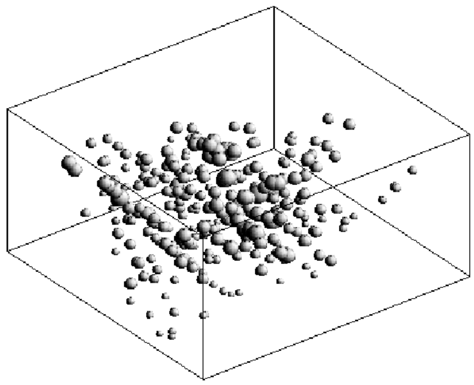


Figure 7: 293 weighted samples (x, y, r) for circle particles in the atomic space ϖ_c.

spaces Ω_l, Ω_c and Ω_Ψ can be computed by the probabilities in the five atomic spaces. We discuss three methods.

Method I: conditional binding. This composes object particles by using the atomic features sequentially. For example, for a line object $\theta = (\rho, \tau, s, e, \mu)$, we compute

$$q(\theta) = q_l(\rho, \tau) q_e(s|\rho, \tau) q_e(e|\rho, \tau, s) q_g(\mu|\rho, \tau, s, e). \quad (5)$$

A set of line segments $\theta_l^{(i)} = (\rho_l^{(i)}, \tau_l^{(i)}, s_l^{(j)}, e_l^{(k)})$ are sampled from $q(\theta)$,

$$\{\theta_l^{(i)} \; : \; i = 1, 2, ..., n_L\}.$$

We call them the *object particles* in Ω_l. In a similar way, we can generate object particles in Ω_c.

These object particles appear, in spirit, similar to hypotheses in engineering methods. However, there is one crucial difference. Each object particle represents a windowed domain in the object space not just one point. The union of the windows of the object particles covers the entire object space. To make the Markov chain reversible, each time we propose a new object by sampling the proposal probability *not just choosing from the set of particles.*

Object particles should also be generated by recursively composing object particles as shown by the arrows in figure 3.

Method II: off-line composition . One may compose a particle by combining two other particles if the latter are *compatible.* The composition occurs off-line, that is, before we start run MCMC. But this is extremely expensive because of the exponential number of possible combinations[3].

Method III: on-line composition. This differs from method II by binding objects during the MCMC computation when two compatible objects appear (or are "alive") in the current W.

Atomic particles are computed once in a bottom-up process, whereas object particles are assembled dynamically during the MCMC process. Figure 8 shows three rows of object particles. One for birth candidates, one for death candidates and decomposition candidates, and one for compatible composition pairs which are alive in W. We must also consider the interactions between objects through the effects of occlusion captured by the $\boldsymbol{\alpha}$ map. For example, suppose a long line segment in the image is covered, at present, by two short line objects $\theta_l^{(i)}$ and $\theta_l^{(j)}$. Then adding the long line object $\theta_l^{(k)}$ will have very little gain in fitting the image because of occlusion effect. Therefore the weights of these object particles must be updated on-line. We refer to a technical report for details[12].

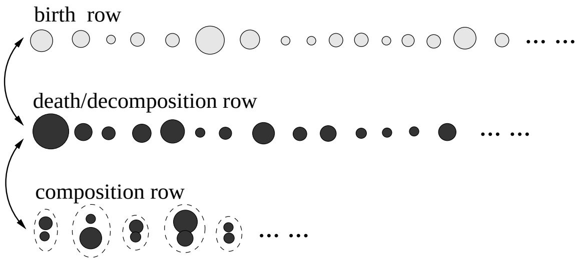


Figure 8: Object particles are organized in three rows which drive the MCMC dynamics. Dark particles are alive.

6 Computing multiple, distinct solutions – scene particles

To build robust vision systems for object recognition, the traditional MAP (*maximum a posteriori*) estimator

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

is not enough. Instead we should sample $p(W|\mathbf{I})$ and compute a set of representative solutions. However, when Ω is complicated and of high dimensions, simply sampling $p(W|\mathbf{I})$ only generates solutions which are all from a single mode and have trivial differences from each other. Therefore a mathematical criterion must be derived for preserving important, distinctive local modes.

Let $S = \{W_1, W_2, ..., W_M\}$ be M solutions with weights $\omega_i \propto p(W|\mathbf{I}), \forall i$. The M weighted samples encode $p(W|\mathbf{I})$ in a non-parametric form.

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

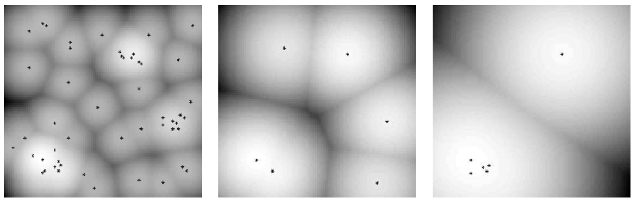
by some Gaussian window function G, we refer to [12] for a distance measure. In this article, we propose a new criterion for extending the traditional MAP.

$$S^* = \{W_1, W_2, ..., W_M\} = \arg\min_{|S|=M} D(p||\hat{p}). \qquad (6)$$

We call $\{W_1, W_2, ..., W_M\}$ the *scene particles*. They are chosen to minimize the Kullback-Leibler divergence $D(p||\hat{p})$ so that $\hat{p}$ "best" preserves p – the true posterior distribution under the constraint of complexity M.

In practice, p is represented as a mixture of Gaussian model like $\hat{p}$ by a large number $N >> M$ of particles recorded during the MCMC process. Thus we choose M distinct solutions during the MCMC computation so that $D(p||\hat{p})$ is minimized.

Figure 9 illustrates this new criterion by plotting $\log p(W)$ in a 2D space. a): $\log p(W)$ with p being a mixture of $N = 50$ Gaussians. The black dots show the centers of the Gaussians. b): $\log \hat{p}(W)$ with $M = 6$ selected particles that minimize $D(p||\hat{p})$. c): $\log \hat{p}(W)$ with $M = 6$ selected particles that minimize $|p - \hat{p}|$ in L1 norm. Obviously the KL-divergence obtains more distinctive and dispersive solutions.



a b c

Figure 9: See text for explanation.

A mathematical deduction of the $D(p||\hat{p})$ leads us to three beautiful principles guiding the selection of scene particles. The proof is available in a technical report[12].

1. One of the particles must be the global optimum W^*. Missing W^* causes substantial increase in $D(p||\hat{p})$.

2. The scene particles should minimize the total sum of energy (or maximize the product of probabilities).

3. The scene particles should also maximize the sum of distances from each other.

The last two are conflicting, and thus the particles must "occupy" distinct modes in the solution space Ω.

7 Experiment I: the Ψ-world

We used an ensemble of randomly generated images for experiments, two of which are shown in figure 1. Figure 10 displays the steps in an MCMC algorithm by showing the $\mathbf{I}_o$ decided by W at step t. Figure 11 displays three different solutions for the input image shown in figure 1.a. Notice that some arcs are represented by a few short line segments, and some are represented by a single arc and so on.

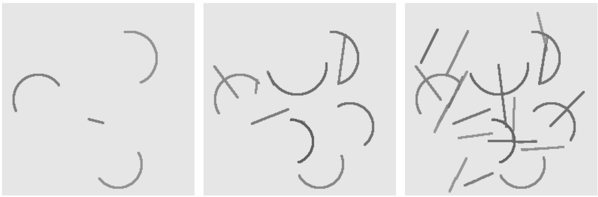


Figure 10: W visited by MCMC at time steps.

Figure 11: Three results of an image in figure 1.

We are mainly interested in studying the improvement of the MCMC efficiency by comparing four Markov chains.

MCMC I: the Markov chain uses uniform proposal probabilities, as in the literature[6, 5].

MCMC II: the Markov chain uses atomic proposal probabilities with Hough transform without end point detection.

MCMC III: the Markov chain uses both Hough transform and end point detection, and samples new objects randomly from the set of particles.

MCMC IV: the Markov chain uses both Hough transform and end point detection. But it is different from MCMC III in that it the weights of object particles are evaluated on-line.

Figure 12 plots the energy level, which is $-\log p(W|\mathbf{I})$, of the MCMC state at step t. Each of the four Markov chains runs 10 times for each of the 8 images, and the energy curves are thus averaged over 80 runs. The dotted, dash-doted, dashed, and solid

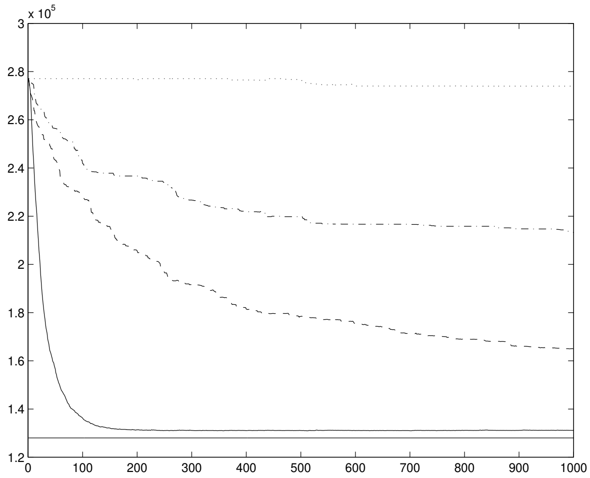


Figure 12: The energy of W plotted against time.

curves are respectively for the MCMC I,II, III, and IV. The horizontal line at the bottom is the averaged energy for the truth W^*s. It is clear that with more data-driven methods used the Markov chains can approach the solution at much faster speed. MCMC IV reaches the solution with 2.3% relative error in 200 steps. The error is mainly a problem of fine-tuning by diffusion. In contrast, MCMC III needs about $5,000$ steps. MCMC I does not go down after even millions of steps. We have also compared the "mix" rate by measuring *how broad* the obtained solutions S are distributed in Ω.

8 Experiment II: traffic sign detection

Figure 13: An input image of street signs.

We applied the same method to traffic sign detection. We used a set of street sign images from TranMap Co., and one is shown in figure 13. We also constructed a set of typical template signs, of which the front views are displayed in figure 14.

The traffic signs are organized in a hierarchy in figure 15. This is similar to the hierarchy studied in[2].

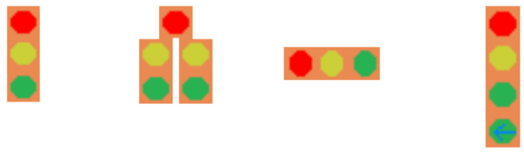


Figure 14: The front views of some templates.

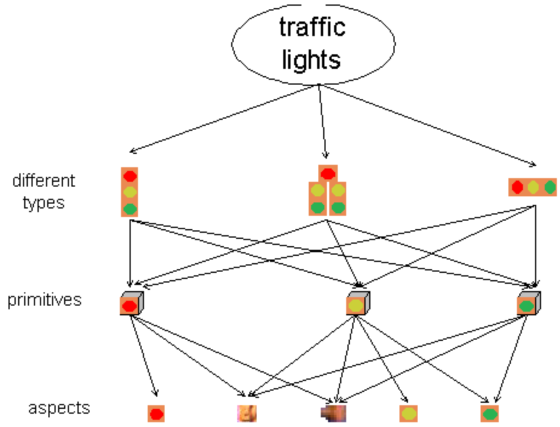


Figure 15: The hierarchy in the sign models.

Firstly, it has different types of signs. Secondly, these signs consists of a set of small primitives with single light. thirdly, the primitives have different aspects (typical views), which are further detectable by color and edge/corner features. Figure 16 (top) shows the object particles (hot spots) of single light primitive computed by the bottom-up processes. The image below is the final result with the top-down template matched to the signs indicated by the two highlighted boundaries.

9 Discussion

We refer to a technical report[12] for the details of three key issues.

1. The models for the background clutter in traffic sign, and how we integrate some simple image segmentation methods with object recognition.

2. The mathematical criterion for what are good feature detectors in computing the importance proposal probabilities $q()$s.

3. The distance, used in keeping distinct solutions, between two solutions $d(W_i, W_j)$ is defined by the minimum "work" needed to change one to the other.

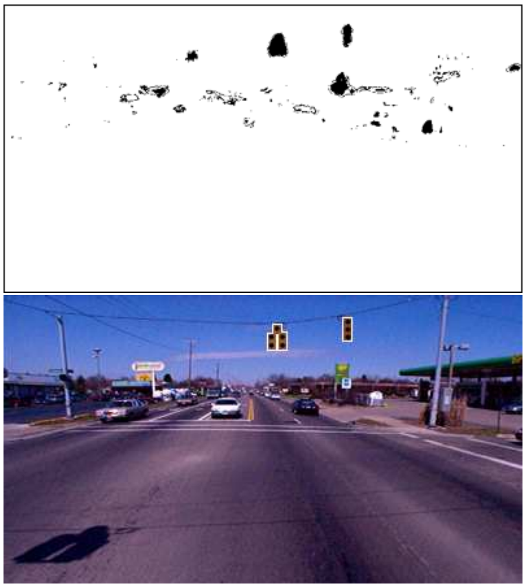


Figure 16:

References

[1] D. Comaniciu and P. Meer, "Mean shift analysis and applications", *Proc. ICCV*, 1999.

[2] S. J. Dickinson, A. P. Pentland, and A. Rosenfeld, "From volumes to views: an approach to 3D object recognition", *CVGIP*, Vol.55, March, 1992.

[3] S. Geman, D. F. Potter, and Z. Chi, "Compositional systems", Preprint of DAM, Brown Univ., 1998.

[4] S. Geman and D. Geman. "Stochastic relaxation ...". *IEE Trans. PAMI* **6**, 1984.

[5] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination", *Biometrika*, vol. 82, 711-732, 1995.

[6] U. Grenander and M. I. Miller, "Representation of knowledge in complex systems", *J. R. Stat. Soc., B*, vol 56, 1994.

[7] M. Isard and A. Blake. "Contour tracking by stochastic propagation of conditional density". *ECCV*, 1996.

[8] S. Loffe and D. Forsyth, "Finding people by sampling", *Proc. ICCV*, 1999.

[9] S. C. Zhu, "Stochastic Jump-diffusion process for computing medial axis in MRFs", *IEEE Trans. PAMI*, vol. 21, no.11, 1999.

[10] S. C. Zhu, Y. N. Wu and D. B. Mumford, "Minimax entropy principle and its application to texture modeling", *Neural Computation* Vol. 9, no 8, Nov. 1997.

[11] S. C. Zhu and Z. Tu, "A symphony algorithm for image segmentation using DDMCMC", TR, OSU Vision And Learning lab., 2000.

[12] S. C. Zhu, R. Zhang, and Q. Luo, "Stochastic computing by Data Driven MCMC in computer vision", TR, OSU Vision And Learning lab., 1999.