Proceedings of Int'l Conf. on Computer Vision, Vancouver, Canada, July, 2001.

Image Segmentation by Data Driven Markov Chain Monte Carlo

Zhuowen Tu, Song-Chun Zhu

Ohio State University
$\{ztu, szhu\}$@cis.ohio-state.edu

Heung-Yeung Shum

Microsoft Research Beijing
hshum@microsoft.com

Abstract [1]

This paper presents a computational paradigm called Data Driven Markov Chain Monte Carlo (DDMCMC) *for image segmentation in the Bayesian statistical framework. The paper contributes to image segmentation in three aspects. Firstly, it designs effective and well balanced Markov Chain dynamics to explore the solution space and makes the split and merge process reversible at a middle level vision formulation. Thus it achieves globally optimal solution independent of initial segmentations. Secondly, instead of computing a single maximum a posteriori solution, it proposes a mathematical principle for computing multiple distinct solutions to incorporates intrinsic ambiguities in image segmentation. A k-adventurers algorithm is proposed for extracting distinct multiple solutions from the Markov chain sequence. Thirdly, it utilizes data-driven (bottom-up) techniques, such as clustering and edge detection, to compute importance proposal probabilities, which effectively drive the Markov chain dynamics and achieve tremendous speedup in comparison to traditional jump-diffusion method[4]. Thus DDMCMC paradigm provides a unifying framework where the role of existing segmentation algorithms, such as, edge detection, clustering, region growing, split-merge, SNAKEs, region competition, are revealed as either realizing Markov chain dynamics or computing importance proposal probabilities. We report some results on color and grey level image segmentation in this paper and refer to a detailed report and a web site for extensive discussion.*

1 Motivation and Introduction

Image segmentation is a long standing problem in computer vision. In recent years there appeared some concerns that image segmentation is perhaps ill-defined, or even a wrong problem to work on, in comparison to other vision tasks which have *apparently well defined* objectives, such as detection, recognition, and tracking. Unfortunately, without addressing segmentation problems, those special purpose vision tasks are fundamentally ill-defined, and have to suffer from generality and robustness. Hard lessens were learned from engineering practices for building large vision systems, for example, automatic target recognition (ATR). One cannot solve a target recognition problem robustly without sufficient modeling and computing for background clutter. Indeed, the confusion in defining image segmentation (and also perceptual grouping) problems simply reflect the following two facts.

I). Computer vision needs to address the issue of formulating the tasks of a general purpose vision system. As David Marr pointed out in his primal sketch, real world images consist of multiple layers of stochastic processes, such as texture, texton, stochastic point, line, curve, graph, region, and object processes, which generate images through spatial organizations. Thus an appropriate formulation should be *image decomposition* or *image parsing* which decomposes an image into its natural constituents as various stochastic processes. This subsumes image segmentation as region process, and naturally integrates object recognition and perceptual organization. The latter deal with point, line, curve, and object processes. Implicit in this formulation is the notion of generative models for image interpretation in contrast to classification and discrimination. Recent progress in statistical learning, e.g.[13] and follow-up work, brought encouraging promise that various stochastic patterns ranging from texture, to point process, to deformable models may well be modeled under a common principle and these models can be learned from large dataset. Thus we believe the segmentation problem is not fundamentally ill-defined, but needs to be extended with richer models.

II). Image segmentation (or decomposition) is indeed *a computing process* and should not be treated as a task by itself. Real images are intrinsically ambiguous, and our perception changes dynamically, even in a very short time duration, depending on our attention. Generally speaking, the more one looks at an image, the more one sees. It seems narrow-minded to think that a segmentation algorithm just outputs one final result. Instead a segmentation (or decomposition) algorithm should realize the intrinsic ambiguities characterized, say, in a Bayesian posterior probability, and outputs multiple distinct solutions dynamically and endlessly

[1] Detailed reports and more results are referred to www.cis.ohio-state.edu/oval/seg.html. This work is supported by an NSF grant IIS 98-77-127 and a NASA grant NAG-13-00039 to S.C. Zhu.

so that these solutions, as samples, "best preserve" the posterior density in a mathematical principle proposed in this paper. For example, the particle filtering (Condensation) algorithm[6] represents such an idea in object tracking. We argue that this should be realized in generic vision system as well.

With the above motivation, we present a stochastic computing paradigm called *data driven Markov chain Monte Carlo* (DDMCMC) for visual inference. In this paper, we focus on image segmentation problem assuming region process only. The DDMCMC paradigm was tested in object recognition in a related paper[14], therefore the compatibility of segmentation and recognition is ensured.

The DDMCMC paradigm for image segmentation proceeds in four steps.

Firstly, we analyze the solution space for segmentation and anatomize it in three levels. Level 1 is *scene space*, where each point represents an interpretation of the scene in terms of a number of regions. Level 2 are *region spaces*, where each point includes variables for describing one region. Level 3 are various *cue spaces* for variables describing the boundaries, intensity models, color models, texture models, and so on. Thus a solution space is a union of many scene spaces of various dimensions, and a scene space is a product of a number of region spaces, and a region space is a product of a number of cue spaces.

Secondly, we design well balanced Markov chains to explore the solution space. We engage two types of dynamics to ensure ergodicity. The first is jump dynamics that simulate *reversible* split and merge, death and birth, and model switching. The second is diffusion dynamics that simulate boundary deformation, region growing, region competition[12], and model adaptation. We *make the split and merge processes reversible* beyond the pixel labeling level and the algorithm is able to explore globally optimal solutions independent of initial segmentation conditions.

Thirdly, data driven techniques, such as edge detection[2] and tracing, data clustering[5] are utilized to compute heuristic information in various cue spaces. The output are sets of weighted *cue particles*, which encode *importance proposal probabilities* in non-parametric forms. These proposal probabilities are composed to form proposal probabilities in region and scene spaces, and thus drastically improve the Markov chain speed.

Fourthly, a mathematical principle is proposed for selecting and pruning a set of *important* and *distinct* scene particles which encode an estimation to the posterior probability. The multiple solutions are computed to minimize a Kullback-Leibler divergence from the estimate posterior to the true posterior. Pruning is crucial for working in a very large solution space, unlike the particle filtering[6].

In summary, the DDMCMC paradigm is about effectively creating, composing, and pruning particles at three levels of the solutions spaces to compute the Bayesian posterior probability dynamically. The DDMCMC paradigm unifies and reveals the roles of some well-known segmentation algorithms. Split and merge, region growing, Snake and balloon[7], region competition[12], and PDEs can be viewed as various MCMC jump-diffusion dynamics after minor modification. Edge detection[2] and clustering[5] compute importance proposal probabilities.

We report some results on both color and grey level image segmentation with textures and refer to a detailed report and a web site for extensive discussion.

2 Problem Formulation

2.1 The structures of solution space

In image segmentation, an image domain Λ is decomposed into n disjoint regions $R_i, i = 1..n$, and $1 \leq n \leq |\Lambda|$. Therefore a segmentation solution is represented by a vector W.

$$W = (n, \{(\Gamma_i, \Theta_i); i = 1, 2, ..., n\}),$$

where Γ_i is the boundary of region R_i and Θ_i an interior description of pixels in R_i. As the number of region is unknown, the solution space is a union of many subspaces of varying dimensions, which we called *scene spaces*,

$$\Omega = \cup_{n=0}^{|\Lambda|} \Omega_o^n, \quad \Omega_o^n = \underbrace{\Omega_R \times \ldots \times \Omega_R}_{n}$$

where Ω_o^n is the subspace with exactly n regions. The *region space* Ω_R is further decomposed into two *cue spaces*: 1. ϖ_Γ: the space for region partitions. 2. ϖ_θ: the space for parameters of region models.

$$\Omega_R = \varpi_\Gamma \times \varpi_\theta.$$

In fact, the region model in ϖ_θ can be further decomposed according to motion, color, texture cues, and each cue is a mixture family of models. We choose not to unfold these details for clarity.

Such space structure is typical, and can become more complicated when we deal with a variety of stochastic patterns as the image decomposition task formulates in section (1).

2.2 Image models

In image segmentation, we are not merely seeking a partition of the image domain, but also a meaningful interpretation of the image in a generative model, in contrast to discrimination model based on feature similarity[11]. Thus we formulate the image segmentation problem in the Bayesian framework,

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

We follow the formulation in region competition[12], and extend it by including richer models obtained in recent statistical learning literature[13].

The prior probability $p(W)$ is the product of three models.

1. A Poisson model for the number of regions $p(n)$.

2. A general smoothness Gibbs prior for the contours $p(\Gamma_i), i = 1..n$.

3. A model of the size of regions $p(A_i)$ where $A_i = size(R_i), i = 1..n$. Recently both empirical and theoretical studies[1, 9] on the statistics of natural images indicates that the size of regions in natural images follows,

$$p(A) \propto \frac{1}{A^{\alpha}}, \quad \alpha \sim 2.0.$$

$p(A)$ in image segmentation encourages large regions to form and empirically plays a non-trivial role in our experiments.

The likelihood model is a product of region models $p(\mathbf{I}_{R_i}|\Gamma_i, \Theta_i)$. $\mathbf{I}_{R_i}$ refers to all pixels jointly inside Γ_i which may include more than one connected components. A region model can be chosen from one of the following four families, and can switch among the four families over time by a Markov chain jump process.

1. A Gaussian model assuming that pixels in a region are independently and identically distributed (iid).

2. A multinomial model assuming that pixels in a region are iid according to a non-parametric intensity histogram.

3. A texture model FRAME [13] with higher order interactions captured by a set of Gabor filters of small size. This model is formulated by a pseudo-likelihood distribution for computational effectiveness.

4. A 2D spline model that account for global inhomogeneous patterns, for example, shadows, shading, sky, perspective textures which are not characterized by models 1-3.

In experiments, we work on both color and grey level images. In color cases, we adopt a (L, U, V) color space and assume Gaussian or mixture of Gaussian models in this color space. We refer to our long report and a web site for detailed account.

We choose these four families of models based on considerations of computational complexity and image contents. We cropped various regions manually from real images and tested our models. Figure 1 shows four typical types of regions that appear most frequently in real world images. For each region $\mathbf{I}_{R_i}$ in the first column (regions $i = 1..4$) , we compute the per-pixel coding bits $L_{ij} = -\log p(\mathbf{I}_{R_i}|\Theta_j^*)$ for models $j = 1..4$ respectively, after fitting the parameters Θ_j^* by a maximum likelihood estimate (MLE) within each family. The smaller L_{ij} is, the better the model j is fit to region i. We then draw a typical sample (synthesis) from $p(\mathbf{I}_R|\Theta_j^*)$ to visualize each model. The synthesized regions are shown in columns 2-5 respective, and the L_{ij}'s are listed below the synthesized images. Obviously better synthesis can be achieved if we can afford computational costs of more sophisticated models. Choosing the right set of models is essential for solving segmentation problems.

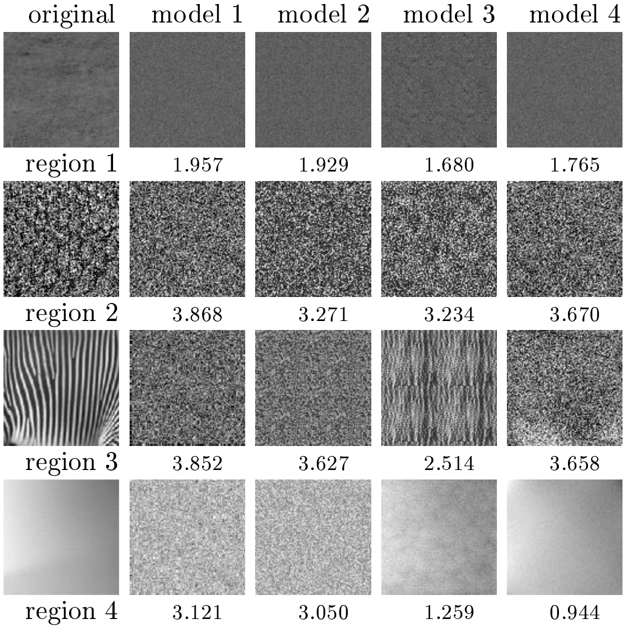


Figure 1: Comparison study of four families of models. See text for explanations.

3 Segmentation by DDMCMC

The posterior probability $p(W|\mathbf{I})$ not only have enormous number of local maxima but is distributed over subspaces of varying dimensions. Searching for a globally optimal solution in such a complicated space is not a trivial task, but still tractable. In the literature, we see three typical works. I). variational methods[1], such as SNAKE/Balloon[7], PDEs[10], region growing, and region competition[12], solve very well a diffusion problem. That is, given a initial segmentation, evolve the boundary to a locally optimal energy state. II). Algorithms involving large operators, such as split and merge, death and birth, model switching were not reversible. III). Algorithms such as clustering and edge detection are effective, but lack theoretical assurance.

The DDMCMC method intents to unify them under a common framework.

3.1 Designing ergodic Markov chain dynamics

The first requirement for designing a global search algorithm is that the Markov chain must be ergodic. That is, starting from an arbitrary initial segmentation, it should be able to visit any other state in the solution space in finite time steps. This disqualifies all greedy algorithms. It requires both jump-dynamics which move between subspaces of varying dimensions and diffusion dynamics which move within a subspace of a fixed dimension. The second requirement is that the Markov chain must have the posterior $p(W|\mathbf{I})$ as its invariant probability at equilibrium. This is often replaced by a strong condition of detailed balance equations which also demands that every move be reversible, in particular for the jumps[3].

We design five types of MCMC dynamics in the following.

Type I: Diffusion of region boundary.

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.

We discuss type I, II, and III briefly in the following, and the other two are designed in similar ways.

Type I evolves the boundary to maximize the posterior probability through a region competition equation[12]. Let Γ_{ij} be the boundary between R_i, R_j $\forall i, j$,

$$\frac{d\Gamma_{ij}(s)}{dt} = (f_{prior}(s) + \log\frac{p(\mathbf{I}(x(s), y(s)|R_i)}{p(\mathbf{I}(x(s), y(s)|R_j)} + \sqrt{2T_t}dB_t)\vec{n}(s)$$

The first two terms are derived from the prior and data directly which are forces along the contour normal $\vec{n}(s)$. dB_t is a Brownian motion and T_t s a temperature parameter. In region competition[12], a small window was used to avoid being stuck in local pitfalls, this in fact can be done through the above stochastic Langevin equations in a principled way.

Types II and III are a pair of jump processes. Suppose at a certain time step t, we propose to split a region R_i into two regions, denoted by R_j and R_k, and thus move the Markov chain from W to W',

$$W = (n, \Theta_i, w) \to (n+1, \Theta_j, \Gamma_{jk}, \Theta_k, w) = W',$$

where w is the remaining variables which do not change during the move. In the Metropolis-Hastings method, we need to have two proposal probabilities $G(W \to dW')$ for the move and $G(W' \to dW)$ for immediately moving back. Then the proposed move is then accepted with probability

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

The split proposal probability is

$$G(W \to dW') = q(II)q(i)q(\Gamma_{jk}|R_i)q(\Theta_j, \Theta_k|\Gamma_{jk}, R_i)dW',$$

where $q(II)$ is the probability for choosing move type II, $q(i)$ is the probability for choosing a region i. $q(i)$ is often decided by the goodness of fit in region R_i, a region with bad fit has a higher chance to split. $q(\Gamma_{jk}|R_i)$ is the probability for selecting a candidate boundary within R_i, and $q(\Theta_j, \Theta_k|\Gamma_{jk}, R_i)$ is the probability for choosing two new models for the new regions. The merge proposal probability is,

$$G(W' \to dW) = q(III)q(j, k)q(\Theta_i)d\Theta_i dw,$$

where $q(III)$ is the probability for choosing type III, $q(j, k)$ and $q(\Theta_i|j, k)$ are the probabilities to choose regions j and k forming region i.

The effectiveness of MCMC depends critically on the design of the proposal probabilities, for example, $q(\Gamma_{jk})$ and $q(\Theta_j, \Theta_k|\Gamma_{jk})$. Naive proposals have lower chance to be accepted and result in exponential waiting time. In the following two subsections, we discuss how importance proposal probabilities are computed using data driven methods.

3.2 Computing importance proposal probabilities in cue spaces

We use a simple color image as a working example and assume only a Gaussian model for color in (L,U,V) space for clarity. In experiments, we do the same importance proposals in each of the four image models.

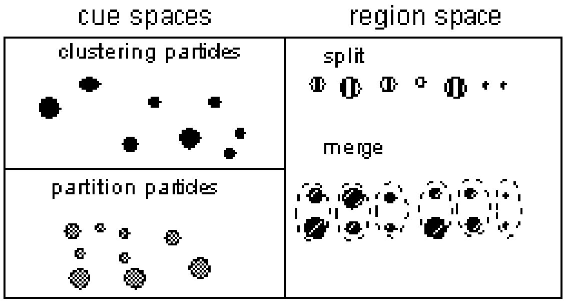
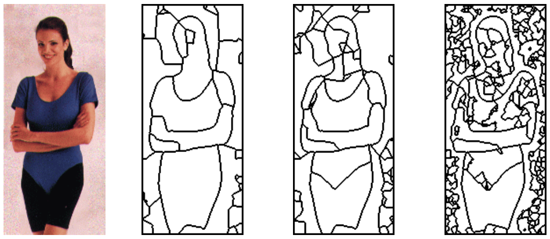


Figure 2: Importance proposal probabilities in two cue spaces and a region space.

The basic ideas of computing importance proposal probabilities are shown in Figure 2. The DDMCMC algorithm starts with some data driven methods for computing a set of weighted particles in each of the cue spaces, which are then composed into particles in the region space. In this example, we have two cue spaces: the boundaries ϖ_Γ and a color ϖ_θ, within which we compute two importance proposal probabilities respectively.

$$\begin{aligned} q_b(\pi) &= \sum_{i}^{n_b} w_b^{(i)} G(\pi - \pi^{(i)}), & \sum_{i=1}^{n_b} w_b^{(i)} = 1, \\ q_c(\theta) &= \sum_{i=1}^{n_c} w_c^{(i)} G(\theta - \theta^{(i)}), & \sum_{i=1}^{n_c} w_c^{(i)} = 1, \end{aligned}$$

where n_c and n_b are the number of weighted particles. $G()$ is a Parzen window function centered at 0. An example is shown in Figure 3 and Figure 4.



a). input b). scale 1 c). scale 2 d). scale 3

Figure 3: Importance proposals for edges at three scales of details. Obtained through edge detection and simple tracing.

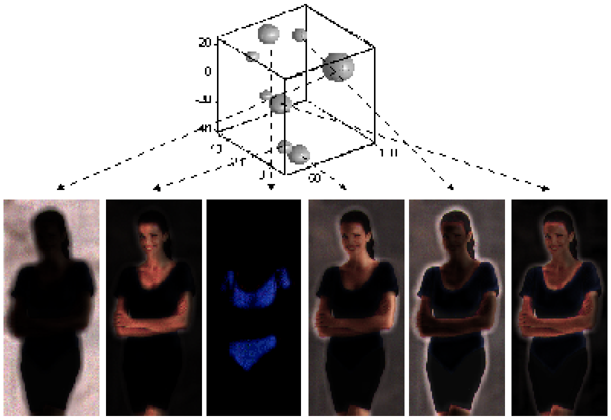


Figure 4: The first row shows color clusters in (L, U, V) space, and the second row are six saliency maps corresponding to the color clusters. (See text)

Figure 3.a shows a color image used in the *region competition* algorithm[12]. Figures 3.b-d show three edge maps obtained by a Canny edge detector followed with a simple edge tracing method to form closed boundaries. The three scales are generated with three levels of thresholds. These edge maps represent a huge number of possible partitions of the image domain, and these partitions are organized in a hierarchy from coarse-to-fine. Each possible partition encodes a number of cue particles in the space ϖ_Γ.

For this simple image, we use an iid Gaussian model for the color θ in (L, U, V) space. Therefore, A mean-shift clustering algorithm[5] is used to compute color clusters in ϖ_θ. Several color clusters are shown in Figure 4 (top row). Each cluster is associated with a "saliency map" in the second row. Each saliency map is obtained by multiplying the original input image by the probability of each pixel belonging to this cluster. This is, the clustering is a soft decision on the pixels. For example, the cluster for the leftmost saliency image in Figure 4 (the second row) is for the background, and the second one is for the skin, and so on.

3.3 Computing importance proposal probabilities in region space

With the cue particles and importance proposal probabilities in the cue spaces, the DDMCMC paradigm proceeds to generate region particles which are large hypotheses in the region space. These region particles are assembled on-line during the Markov chain Monte Carlo computing. It helps to imagine that they are organized into various lists for the different jump-dynamics, such as model switching, split and merge, as the right panel of Figure 2 shows.

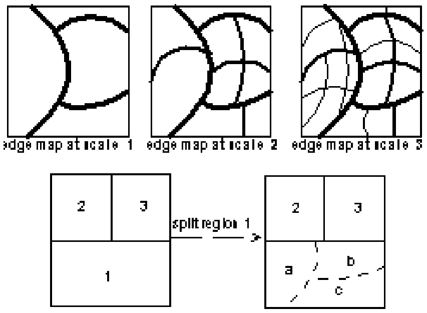


Figure 5: The proposal for splitting a region 1.

We show one example for generating a split proposal by revisiting $G(W \to W')$ in the MCMC design subsection3.1,

$$G(W \to dW') = q(II)q(i)q(\Gamma_{jk}|R_i)q(\Theta_j, \Theta_k|\Gamma_{jk}, R_i)dW',$$

Once a region R_i is proposed to be split, for example, region R_1 in Figure 5. The algorithm randomly, with probability $p^{(s)}$, picks one out of three partition maps generated by edge-based segmentation at different scales. For example, it may choose the first scale in Figure 5 (the first row). The proposed region R_1 is then superimposed on this partition map. The boundaries in R_1 suggest a number of sub-regions, for example sub-regions a, b, and c in Figure 5. The algorithm chooses 1 region from all sub-regions, or it may choose a combination of 2, or 3 regions in case there are many sub-regions. This proposes a boundary to split R_1.

$$q(\Gamma_{jk}|R_i) = \sum_{s=1}^{3} p^{(s)} p_{\Gamma}^{(s)}(R_j, R_k|R_i)$$

where $p_{\Gamma}^{(s)}(R_j, R_k|R_i)$ is the probability for splitting region R_i at the partition map $s \in \{1, 2, 3\}$ into two regions j and k, computed through a number of combinations. We allow a region to consist of multiple disconnected sub-regions to account for occlusion effects.

Given the proposed Γ_{jk}, we need to propose two new models for the new regions R_j, R_k respectively by $q(\Theta_j, \Theta_k|\Gamma_{jk}, R_i) = q(\Theta_j|\Gamma_{jk}, R_i)q(\Theta_k|\Gamma_{jk}, R_i)$. To compute $q(\Theta_j|\Gamma_{jk}, R_i)$, all the color cue particles (clusters) accumulate a vote from all pixels (or a subsampled set) in R_j. The vote received from a pixel is the probability in the saliency map corresponding to this cluster.

$$q(\Theta_j|\Gamma_{jk}, R_i) = \frac{1}{\gamma} \sum_{m=1}^{n_c} \gamma_m G(\Theta_j - \Theta_m), \quad \gamma_m = \sum_{(x,y)\in R_j} \mathrm{Smap}_m(x, y)$$

where γ is the normalizing constant, and Smap_m is the saliency map corresponding to the m-th cluster. The same is done with $q(\Theta_k|\Gamma_{jk}, R_i)$. This concludes the split proposal. The merge proposal will follow other heuristics, such as region similarity etc. We refer to our report for details.

3.4 Computing multiple distinct solutions — scene particles

As we argued at the beginning, image segmentation is a process not a task. We shall compute multiple distinct solutions from the posterior. Simply sampling $p(W|\mathbf{I})$ only generates solutions which are all from a single mode with trivial differences. In this section, we propose a mathematical criterion for preserving important and distinctive solutions.

Let $S = \{(\omega_i, W_i) : i = 1..K\}$ be a set of K solutions (or scene particles) with weights $\omega_i \propto p(W|\mathbf{I}), \forall i$. S encodes a probability in a non-parametric form,

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

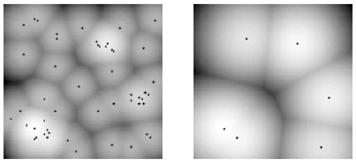
by some Gaussian window function G. To best preserve the posterior probability, we propose to compute S so that $\hat{p}(W|\mathbf{I})$ approaches $p(W|\mathbf{I})$ in terms of minimizing a Kullback-Leibler divergence $D(p||\hat{p})$ under the constraint that $|S| = K$,

$$S^* = \arg\min_{|S_K|=K} \int p(W|\mathbf{I}) \log \frac{p(W|\mathbf{I})}{\hat{p}(W|\mathbf{I})} dW. \qquad (1)$$

In practice, we can always express $p(W|\mathbf{I})$ by a mixture of Gaussian models like $\hat{p}$, but just use a large number $N >> K$ of particles $\Pi = \{(\alpha_j, \pi_j) : j = 1..N\}$ which are, theoretically available, through MCMC simulation.

$$p(W|\mathbf{I}) = \sum_{j=1}^{N} \frac{\alpha_j}{\alpha} G(W - \pi_j), \quad \sum_{j=1}^{N} \alpha_j = \alpha.$$

We denote the scene particles by π_j with weight α_j to avoid confusion with the particles in S. Thus the problem becomes to select and prune S from Π during the MCMC sampling process.



a). $N = 50$ b). $K = 6$

Figure 6: a). $\log p$ with $N = 50$ particles, b). $\log \hat{p}$ with $K = 6$ particles.

This idea is illustrated in Fig. 6. Fig. 6.a shows $\log p$ for a distribution with $N = 50$ Gaussians, which is approximated by $\hat{p}$ with $K = 6$ Gaussians in Fig. 6.b through minimizing $D(p||\hat{p})$.

For two mixtures of Gaussians, some simple expression is available for $D(p||\hat{p})$ under reasonable conditions and simplifications.

Proposition In the above notations, with mild assumptions for the posterior probability, we have $D(p||\hat{p})$ approximated by

$$log\frac{\alpha}{\omega} + \sum_{j=1}^{N} \frac{\alpha_j}{\alpha}[\log \frac{\alpha_j}{\omega_{c(j)}} + \frac{|\pi_j - W_{c(j)}|^2}{2\sigma^2}]. \qquad (2)$$

$c : \{1..n\} \rightarrow \{1..K\}$ is a mapping of index. $W_c(j)$ is the closest particle in S to π_j in Π.

Equation (2) has very intuitive meanings. Because the probability sums α and ω in both Π and S are dominated by the best particles in the sets respectively (because they are exponentially distributed), the first term suggests that the globally optimal solution should be included in S. This proves that eqn (1) generalizes the traditional (*maximum a posteriori*) estimator $W^* = \arg\max_{W\in\Omega} p(W|\mathbf{I})$ as $K = 1$. The second term suggests that each selected scene particle W should have large weight while the third term keeps the particles in S apart from each other. In the solution space, the distance between two solutions $|\pi_j - W_{c(j)}|$ is defined to reflect some perceptual differences, for example, the minimum work to move π_j to $W_{c(j)}$.

We use a greedy algorithm to minimize $D(p||\hat{p})$, which we called the *K-adventurer* algorithm. Suppose we have a set of K particles S at step t. At time $t+1$, we obtain new particles by MCMC. We augment set S to S_+ by including these new particles. Then we treat S_+ as Π and choose K particles by eliminating some particles in S_+. The algorithm updates the particles in the set stepwisely to minimize the divergence.

4 Experiments

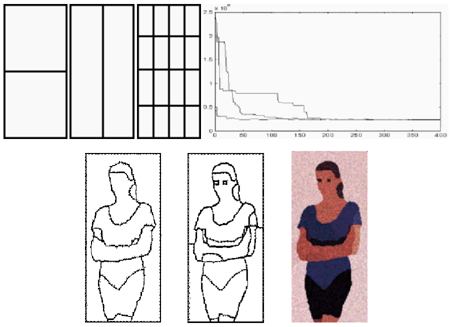


Figure 7: Simulating three Markov Chains with different initial states.

The DDMCMC paradigm was tested extensively on many grey level, color, textured images, and this section shows a few examples. More are available in our web site.

We first show an simple working example to illustrate the DDMCMC steps. Following the importance proposal probabilities for the edges in Figure 3 and for color clustering in Figure 4, we simulated three Markov chains with three different initial segmentation shown in Figure 7 (top row). The energy changes ($-\log p(W|\mathbf{I}^{\text{obs}})$) of the three MCMCs are plotted in Figure 7 against time steps. All Markov chains converge very quickly to a nice solution in a few minutes on a Pentium III PC. Figure 7 shows two different solutions obtained by a Markov chain. To verify the computed solution W^*, we synthesized an image by sampling from the likelihood $p(\mathbf{I}|W^*)$. Obviously, some highlight color and facial expression are missing as they are not included in the models. This is also a way to exam the sufficiency of models in segmentation (see section (2)).

Figure 10 shows some multiple segmentation results on a gray-scale zebra image. The results are obtained by the K-adventurers algorithm. Note that these results are very intuitive and capture distinct interpretations of the image.

Figure 8 and Figure 9 displays some other color and grey level images using the same algorithm. We show the input (left) and a segmentation (middle) starting with arbitrary initial conditions and a synthesized image (right) drawn from the likelihood $\mathbf{I}^{\text{syn}} \sim p(\mathbf{I}|W)$. Obviously the segmentation results will improve if we integrate other stochastic models. For example, in the first row of Figure 8. The music band in a football stadium forms a point process some other images may have line processes. Including these patterns will extend the system from image segmentation to image decomposition and parsing.

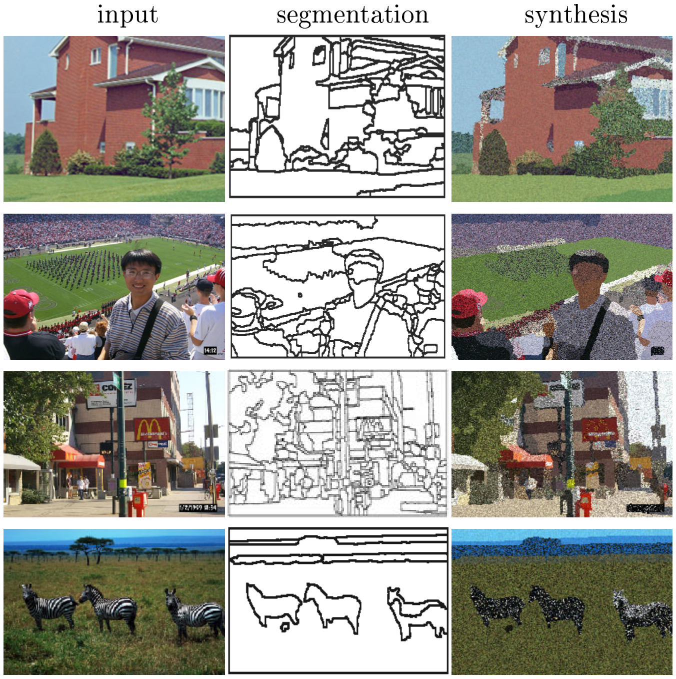
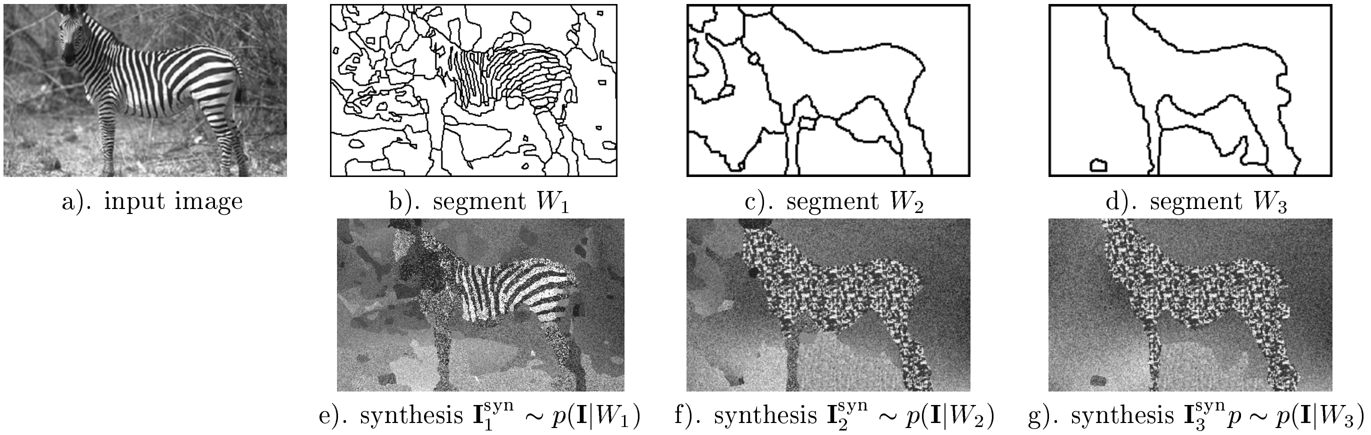


Figure 8: Color image segmentation by DDMCMC. Left: input images, middle: segmentation results W, right: synthesized images $\mathbf{I}^{\text{syn}} \sim p(\mathbf{I}|W)$ with the segmentation results w.

In our final example, we examed the sensitivity of DDMCMC to three conditions: 1). the initial segmentation, 2). the importance proposal at ϖ_Γ, 3). the importance proposal at ϖ_θ. In figure 7 we can see that the selection of different initial states makes difference at burn-in period. Three MCMCs quickly converge together after a few hundred iterations which is a small portion in the overall time spent by MCMC processes that requires several thousand iterations to converge.



a). input image b). segment W_1 c). segment W_2 d). segment W_3

e). synthesis $\mathbf{I}_1^{\text{syn}} \sim p(\mathbf{I}|W_1)$ f). synthesis $\mathbf{I}_2^{\text{syn}} \sim p(\mathbf{I}|W_2)$ g). synthesis $\mathbf{I}_3^{\text{syn}}p \sim p(\mathbf{I}|W_3)$

Figure 10: Experiments on a gray-level zebra image with three solutions, e.g., the background is segmented into a couple of regions with spline models which capture the global shading effects and the zebra is a texture model. The synthesized images could be improved if more filters are used.

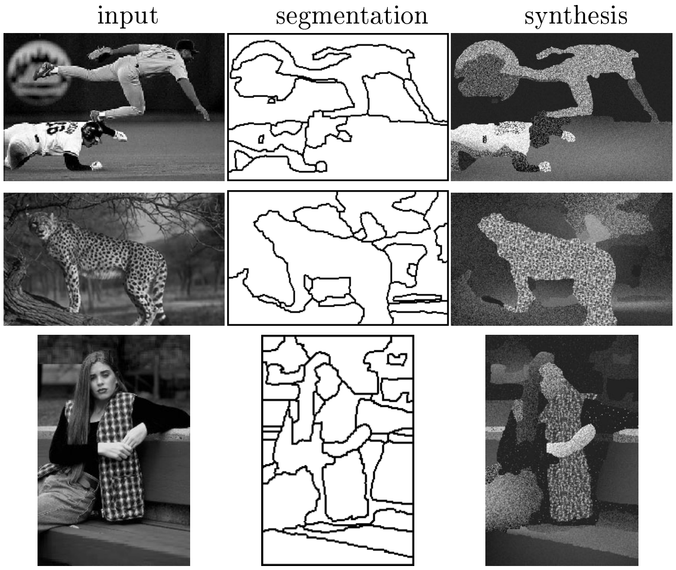


Figure 9: Gray level image segmentation by DDMCMC. Left: input images, middle: segmentation results W, right: synthesized images $\mathbf{I}^{\text{syn}} \sim p(\mathbf{I}|W)$ with the segmentation results w.

5 Discussion

In future work, we should look for analytic results that link the DDMCMC convergence rate to the goodness of the set of heuristics. We should also integrate the segmentation algorithm with other stochastic models, such as texture, point, curve processes, and object recognition[14] under the same computational paradigm.

References

[1] L. Alvarez, Y. Gousseau, J.M. Morel, “The size of objects in natural images”, Preprint of Centre de Math. and Applications, ENS Cachan, Fr. 2000.

[2] J.F. Canny. “A computational approach to edge detection”, *PAMI.* 8(6), 1986.

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

[4] U. Grenander and M.I. Millter, “Representations of Knowledge in Complex Systems”, *Journal of the Royal Stat. Soc. Series B*, vol. 56, isuue 4, 1994.

[5] D. Comaniciu and P. Meer, “Mean shift analysis and applications”, *Proc. ICCV*, 1999.

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

[7] M. Kass, A. Witkin and D. Terzopoulos. “Snakes: Active Contour Models”, *IJCV*, 1, 1988.

[8] G. Koepfler, C. Lopez and J.M. Morel. “A Multiscale Algorithm for Image Segmentation by Variational Approach”. *SIAM J. NUMER. ANAL.* 31(1), 1994.

[9] A. B. Lee, j.G. Huang, and D.B. Mumford, “Random collage model for natural images”, *IJCV*, to appear.

[10] N. Paragios and R. Deriche, “Coupled Geodesic Active Regions for Image Segmentation: A Level Set Approach”, *ECCV*, 2000.

[11] J. Shi and J. Malik, “Normalize cuts and image segmentation”, *CVPR*, 1997.

[12] S. C. Zhu and A. Yuille.“Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multi-band Image Segmentation”, *PAMI.* vol. 18, no.11, 1996.

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

[14] S. C. Zhu, R. Zhang, and Z. Tu. “Integrating Topdown/Bottom-up for Object Recognition”, *CVPR*, 2000.