.

Cluster Sampling and Its Applications in Image Analysis

Adrian Barbu[2] and Song-Chun Zhu[1,2]

8125 Math Science Bldg, Box 951554

Departments of Statistics[1] and Computer Science[2]

University of California, Los Angeles

abarbu@ucla.edu, sczhu@stat.ucla.edu

This manuscript is Preprint 409, Department of Statistics. Some portions of the work were presented in two vision conferences ICCV03 and CVPR04. We acknowledge the support of NSF grant IIS-02-44763. and thank Dr. Yingnian Wu and Dr. David Mumford for stimulating discussions.

Abstract

Markov chain Monte Carlo (MCMC) methods have been used in many fields (physics, chemistry, biology, and computer science) for simulation, inference, and optimization. The essence of these methods is to simulate a Markov chain whose state $\mathbf{X}$ follows a target probability $\mathbf{X} \sim \pi(\mathbf{X})$. In many applications, $\pi(\mathbf{X})$ is defined on a graph $\mathbf{G}$ whose vertices represent elements in the system and whose edges represent the connectivity of the elements. $\mathbf{X}$ is a vector of variables on the vertices which often take discrete values called labels or colors. Designing rapid mixing Markov chain is a challenging task when the variables in the graph are strongly coupled. Methods, like the single-site Gibbs sampler, often experience long waiting time. A well-celebrated algorithm for sampling on graphs is the Swendsen-Wang (1987) (SW) method. The SW method finds a cluster of vertices as a connected component after turning off some edges probabilistically, and flips the color of the cluster as a whole. It is shown to mix rapidly under certain conditions. Unfortunately, the SW method is only applicable to the Ising/Potts models and slow down critically in the presence of "external fields" i.e. likelihood in Bayesian inference.
In this paper, we present a general cluster sampling method which achieves the following objectives. Firstly, it extends the SW algorithm to general Bayesian inference on graphs. Especially we focus a number of image analysis problems where the graph sizes are in the order of $O(10^3) - O(10^6)$ with $O(1)$ connectivity. Secondly, the edge probability for clustering the vertices are discriminative probabilities computed from data. Empirically such data driven clustering leads to much improved efficiency. Thirdly, we present a generalized Gibbs sampler which samples the color of a cluster according to a conditional probability (like the Gibbs sampler) weighted by a product of edge probabilities. Fourthly, we design the algorithm to work on multi-grid and multi-level graphs. The algorithm is tested on typical problems in image analysis, such as image segmentation and motion analysis, In our experiments, the algorithm is $O(10^2)$ orders faster than the single-site Gibbs sampler. In the literature, there are several ways for interpreting the SW-method which leads to various analyses or generalizations, including random cluster model (RCM), data augmentation, slice sampling, and partial decoupling. We take a different route by interpreting SW as a Metropolis-Hastings step with auxiliary variables for proposing the moves or we can view it as a generalized hit-and-run method.

Keywords: Swendsen-Wang, Data Augmentation, Slice sampling, multi-grid sampling, multi-level sampling.

1 Introduction

Markov chain Monte Carlo (MCMC) methods are general computing tools for simulation, inference, and optimization in many fields. The essence of MCMC is to design a Markov chain whose transition kernel $\mathcal{K}$ has an unique invariant (target) probability $\pi(\mathbf{X})$ predefined in a task. For example, $\pi(\mathbf{X})$ could be a Bayesian posterior probability or a probability governing the states of a physical system. In this paper, we are interested in Markov chains with finite states $\mathbf{X}$ defined on graphs $\mathbf{G} =< V, E >$ where $\mathbf{X} = (x_1, x_2, ..., x_n)$ represents the states of the vertices $V = \{v_1, v_2, ..., v_n\}$. Such problems are often referred as graph coloring (or labeling) and have very broad applications in physics, biology, and computer science.

Although the method presented in this paper is applicable to generally graphs and target probabilities, we shall focus on a number of examples in image analysis, such as image segmentation and motion analysis. For such applications, the graph $\mathbf{G}$ is very large with $O(10^4)-O(10^6)$ vertices which are image elements like pixels, and $\mathbf{G}$ has sparse nearest neighbor connections, i.e. constant $O(1)$ connectivity. That is, the connectivity of a vertex does not grow with the number of vertices. The state x_i is the color (or label) for image segmentation or discretized motion velocity in motion analysis. The target probability $\pi(\mathbf{X})$ are usually Markov random fields whose conditional probabilities can be computed locally.

In the literature, a generally applicable algorithm for Markov chain design is the Gibbs sampler (Geman and Geman 1984) and its generalizations, such as multigrid (Goodman and Sokal 1989), parameter expansion (Liu and Wu 1999), parallel tempering (Geyer and Thompson 1995). The slow mixing of such methods is attributed to the strong coupling between variables in the graph. One well celebrated algorithm is the Swendsen-Wang (1987) method designed for simulating the Ising/Potts models (Ising 1925, Potts 1953) in statistical physics. It is often called the cluster sampling method. In each iteration, the SW method forms a cluster of vertices as a connected component by sampling Bernoulli variables defined on each edge. Then it flips the color of all vertices in the cluster simultaneously.

The SW method is found to mix rapidly under certain conditions. For example, Cooper and Frieze (1999) shows that SW has polynomial mixing time for graphs with $O(1)$ connectivity, such as the Ising/Potts models even at near critical temperature. Gore and Sinclair (1999) showed that SW has exponential mixing time when $\mathbf{G}$ is a complete graph. Huber (2002) designed bounding chains for the SW method so as to diagnose exact sampling in some temperature range of the Potts model (see Fig. 2). The SW convergence can also be analyzed with a maximal correlation technique (Liu 2001, chapter 7). Despite its success, the power of the SW method and its analyses are very limited for two reasons.

1. It is only applicable to the Ising/Potts models and cannot be applied to arbitrary probabilities on general graphs.

2. It uses constant probability for the binary variables on edges, and does not make use of the data information in clustering the vertices. Thus it slows critically in the presence of "external fields" (i.e. data).

In this paper, we present a general cluster sampling algorithm which extends the SW-method in the following aspects.

1. Designed from the Metropolis-Hastings perspective, it is applicable to general probabilities on graphs.

2. It utilizes discriminative probabilities computed from the input data on the edges for compatibilities of the two adjacent vertices. Therefore the clustering step is informed by the data (external field) and leads to significant speedup empirically.

3. In a modified version, it can be viewed as a generalized Gibbs sampler which samples the color of a cluster according to a conditional probability (like the Gibbs sampler) weighted by a product of a small number of edge probabilities. This can also be viewed as a generalized hit-and-run method.

4. It is extended to multi-grid and multi-level graphs for hierarchic graph labeling.

In our two sets of experiments on image analysis (segmentation and motion), the algorithm is $O(10^2)$ times faster than the single-site Gibbs sampler (see Figs.8, 9, and 10).

In the literature, there are two famous interpretations of the SW-method which leads to various analyses or generalizations. Both view the SW method as a data augmentation method (Tanner and Wong 1987).

1. The first is the Random Cluster Model (RCM) by Edwards and Sokal (1988). It augments the target probability $\pi(\mathbf{X})$ with a new set of binary variables $\mathbf{U}$ on the edges. The joint probability $p_{\text{ES}}(\mathbf{X}, \mathbf{U})$ has a marginal probability $\pi(\mathbf{X})$ and two conditional probabilities $p_{\text{ES}}(\mathbf{X}|\mathbf{U})$ and $p_{\text{ES}}(\mathbf{U}|\mathbf{X})$ which are easy to sample. In this model, the clustering and labeling are decoupled completely. It leads to the design of bounding chain (Huber 2002) for exact sampling.

2. The second is the slice sampling and decoupling method by (Higdon 1996). It augments $\pi(\mathbf{X})$ by a set of continuous variables W as the "bond strength" on edges to decouple the internal fields with the external fields, and thus sample the labels under the constraints of these variables (i.e. slice sampling). Higdon applied this method to some image analysis examples and also studied a partial decoupling method which has a coupling factor controlled by the data.

In this paper, we take a third route by interpreting SW as a Metropolis-Hastings step with auxiliary variables for proposing the moves. Each step is a reversible jump (Green 1995) and observes the detailed balance equations. The key observation is that the proposal probability ratio can be calculated neatly as a ratio of products of probabilities on a small number of edges on the border of the cluster.

The paper is organized in the following. We start with a background introduction on the Potts model, SW, and two interpretations in Section (2). Then we derive a generalized method by the Metropolis-Hastings perspective in Section (3). A number of variant methods are presented in Section (4), including the cluster Gibbs sampler and the multiple flipping scheme. Section (5) shows the first experiment on image segmentation. Then we proceed to the multi-grid and multi-level cluster sampling in Section (6). The motion experiments are reported in Section (7). We will compare the design of our method with the single site Gibbs sampler. The paper is concluded in Section (8) with discussions.

2 Background: Potts, SW, and interpretations

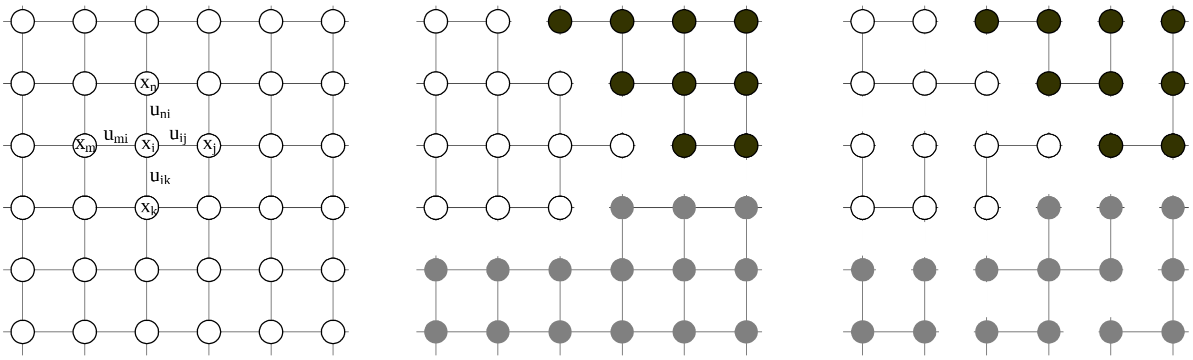
In this section, we review the Potts model, SW method and its two interpretations. The review is made concrete enough so that important results can be followed.

2.1 SW on Potts model

Let $\mathbf{G} =< V, E >$ be an adjacency graph, such as a lattice with 4 nearest neighbor connections. Each vertex $v_i \in V$ has a state variable x_i with finite number of labels (or colors), $x_i \in \{1, 2, ..., \mathrm{L}\}$. The total number of label L is pre-defined, and the Potts model for a homogeneous Markov field is,

$$\pi_{\mathrm{PTS}}(\mathbf{X}) = \frac{1}{Z} \exp\{\beta \sum_{<i,j>\in E} \mathbf{1}(x_i = x_j)\}. \tag{1}$$

$\mathbf{1}(x_i = x_j)$ is a Boolean function. It is equal to 1 if its condition $x_i = x_j$ is observed, and is 0 otherwise. In more general cases, $\beta = \beta(v_i, v_j)$ may be position dependent. Usually we consider $\beta > 0$ for a ferro-magnetic system which prefers same colors for neighboring vertices. The Potts models and its extensions are used as *a priori* probabilities in many Bayesian inference tasks.



(a) G_0

(b) G

(c) CCP

Figure 1: Illustating the SW method. (a) An adjacency graph **G** and each edge $< i, j >$ is augmented with a binary variable $\mu_{ij} \in \{1, 0\}$. (b) A labeling of the Graph **G** where the edges connecting vertices of different colors are removed. (c). A number of connected component after turning off some edges in (b) probabilistically.

As Fig.1.(a) illustrates, the SW method introduces a set of auxiliary variables on the edges.

$$\mathbf{U} = \{\mu_{ij} : \ \mu_{ij} \in \{0, 1\}, \ \forall < i, j > \in E\}. \tag{2}$$

The edge $< i, j >$ is disconnected (or turned off) if and only if $\mu_{ij} = 0$. μ_{ij} follows a Bernoulli distribution conditioning on x_i, x_j.

$$\mu_{ij}|(x_i, x_j) \sim \text{Bernoulli}(\rho \mathbf{1}(x_i = x_j)), \quad \rho = 1 - e^{-\beta}. \tag{3}$$

$\mu_{ij} = 1$ with probability ρ if $x_i = x_j$, and $\mu_{ij} = 1$ with probability 0 if $x_i \neq x_j$. The SW method iterates two steps.

1. The clustering step. Given the current state $\mathbf{X}$, it samples the auxiliary variables in $\mathbf{U}$ according to eqn. (3). It first turns off all edges $< i, j >$ deterministically if $x_i \neq x_j$, as Fig.1.(b) shows. Then it turns off the remain edges with probability ρ. The edge $< i, j >$ is divided into the "on" and "off" sets respectively depending on $\mu_{ij} = 1$ or 0.

$$E = E_{\text{on}}(\mathbf{U}) \cup E_{\text{off}}(\mathbf{U}). \tag{4}$$

The edges in $E_{\text{on}}(\mathbf{U})$ form a number of connected components shown in Fig. 1.(c). We denote all the connected components given $E_{\text{on}}(\mathbf{U})$ by,

$$\text{CP}(\mathbf{U}) = \{\text{cp}_i : \ i = 1, 2, ..., K, \ \text{with} \sum_{i=1}^{K} \text{cp}_i = V\}. \tag{5}$$

Vertices in each connected component cp_i have the same color.

2. The flipping step. It selects one connected component cp $\in$ CP at random and assign a common color y to all vertices in cp. y follows a uniform probability,

$$x_i = y \ \forall v_i \in cp, \quad y \sim \text{unif}\{1, 2, ..., \text{L}\}. \tag{6}$$

In this step, one may choose to repeat the random color flipping for all the connected components in $\text{CP}(\mathbf{U})$ independently, as they are decoupled given the edges in $E_{\text{on}}(\mathbf{U})$.

In one modified version by Wolff (1989), one may choose a vertex $v \in V$ and grow a connected component following the Bernoulli trials on edges around v. This saves some computation in the clustering step, and thus bigger components have higher chance to be selected.

2.2 SW Interpretation 1: data augmentation and RCM

The SW method described above is far from what was presented in the original paper (Swendsen and Wang 1987). Instead our description follows the interpretation by Edward and Sokal (1988), who augmented the Potts model to a joint probability for both $\mathbf{X}$ and $\mathbf{U}$,

$$\begin{aligned} p_{\text{ES}}(\mathbf{X},\mathbf{U}) &= \frac{1}{Z}\prod_{<i,j>\in E}[(1-\rho)\mathbf{1}(\mu_{ij}=0)+\rho\mathbf{1}(\mu_{ij}=1)\cdot\mathbf{1}(x_i=x_j)] &(7)\\ &= \frac{1}{Z}[(1-\rho)^{|E_{\text{off}}(\mathbf{U})|}\cdot\rho^{E_{\text{on}}(\mathbf{U})}]\cdot\prod_{<i,j>\in E_{\text{on}}(\mathbf{U})}\mathbf{1}(x_i=x_j). &(8)\end{aligned}$$

The second factor $\prod_{<i,j>\in E_{\text{on}}(\mathbf{U})}\mathbf{1}(x_i=x_j)$ is in fact a hard constraint on $\mathbf{X}$ and $\mathbf{U}$. Let the space of $\mathbf{X}$ be

$$\Omega=\{1,2,...,\text{L}\}^{|V|}. \quad (9)$$

Under this hard constraint, the labeling $\mathbf{X}$ is reduced to a quotient space $\frac{\Omega}{\text{CP}(\mathbf{U})}$ where each connected component must have the same label,

$$\prod_{<i,j>\in E_{\text{on}}(\mathbf{U})}\mathbf{1}(x_i=x_j)=\mathbf{1}(\mathbf{X}\in\frac{\Omega}{\text{CP}(\mathbf{U})}). \quad (10)$$

The joint probability $p_{\text{ES}}(\mathbf{X},\mathbf{U})$ observes two nice properties, and both are easy to verify.

Proposition 1 *The Potts model is a marginal probability of the joint probability,*

$$\sum_{\mathbf{U}}p_{\text{ES}}(\mathbf{X},\mathbf{U})=\pi_{\text{PTS}}(\mathbf{X}). \quad (11)$$

The other marginal probability is the random cluster model π_{RCM}*,*

$$\sum_{\mathbf{X}}p_{\text{ES}}(\mathbf{X},\mathbf{U})=\pi_{\text{RCM}}(\mathbf{U})=\frac{1}{Z}(1-\rho)^{|E_{\text{off}}(\mathbf{U})|}\cdot\rho^{E_{\text{on}}(\mathbf{U})}\text{L}^{|\text{CP}(\mathbf{U})|}. \quad (12)$$

Proposition 2 *The conditional probabilities of* $p_{\text{ES}}(\mathbf{X},\mathbf{U})$ *are*

$$\begin{aligned} p_{\text{ES}}(\mathbf{U}|\mathbf{X}) &= \prod_{<i,j>\in E}p(\mu_{ij}|x_i,x_j), \quad \text{with } p(\mu_{ij}|x_i,x_j)=\text{Bernoulli}(\rho\mathbf{1}(x_i=x_j)), &(13)\\ p_{\text{ES}}(\mathbf{X}|\mathbf{U}) &= \text{unif}[\frac{\Omega}{\text{CP}(\mathbf{U})}]=(\frac{1}{\text{L}})^{|\text{CP}(\mathbf{U})|} \text{ for } \mathbf{X}\in\frac{\Omega}{\text{CP}(\mathbf{U})}; \ =0 \text{ otherwise.} &(14)\end{aligned}$$

Therefore the two SW steps can be viewed as sampling the two conditional probabilities.

1. Clustering step: $\mathbf{U} \sim p_{\mathrm{ES}}(\mathbf{U}|\mathbf{X})$, i.e. $\mu_{ij}|(x_i, x_j) \sim \text{Bernoulli}(\rho\mathbf{1}(x_i = x_j))$.
2. Flipping step: $\mathbf{X} \sim p_{\mathrm{ES}}(\mathbf{U}|\mathbf{X})$, i.e. $\mathbf{X}(\mathrm{cp}_i) \sim \text{Unif}\{1, 2, ..., \mathrm{L}\}$, $\forall \mathrm{cp}_i \in \mathrm{CP}(\mathbf{U})$.

As $(\mathbf{X}, \mathbf{U}) \sim p_{\mathrm{ES}}(\mathbf{X}, \mathbf{U})$, discarding the auxiliary variables $\mathbf{U}$, we have $\mathbf{X}$ following the marginal of $p_{\mathrm{ES}}(\mathbf{X}, \mathbf{U})$. The goal is achieved,

$$\mathbf{X} \sim \pi_{\mathrm{PTS}}(\mathbf{X}). \tag{15}$$

The beauty of this data augmentation method (Tanner and Wong 1987) is that the labeling of the connected components are completely decoupled (independent) given the auxiliary variables. As $\rho = 1 - e^{-\beta}$, it tends to choose smaller clusters if the temperature ($T \propto \frac{1}{\beta}$) in the Potts model is high, and in low temperature it chooses large clusters. So it can overcome the coupling problem with single site Gibbs sampler.

2.3 Some theoretical results

Let the Markov chain have kernel $\mathcal{K}$ and initial state $\mathbf{X}_o$, in t steps the Markov chain state follows probability $p_t = \delta(\mathbf{X} - \mathbf{X}_o)\mathcal{K}^t$ where $\delta(\mathbf{X} - \mathbf{X}_o)$ (for $\delta(\mathbf{X} - \mathbf{X}_o) = 1$ for $\mathbf{X} = \mathbf{X}_o$ and 0 otherwise) is the initial probability. The convergence of the Markov chain is often measured by the total variation

$$||p_t - \pi||_{\mathrm{TV}} = \frac{1}{2}\sum_{\mathbf{X}} |p_t(\mathbf{X}) - \pi(\mathbf{X})|. \tag{16}$$

The mixing time of the Markov chain is defined by

$$\tau = \max_{\mathbf{X}_o} \min\{t : \ ||p_t - \pi||_{\mathrm{TV}} \leq \epsilon\}. \tag{17}$$

τ is a function of ϵ and the graph compexlity $M = |\mathbf{G}_o|$ in terms of the number of vertices and connectivity. The Markov chain is said to mix rapidly if $\tau(M)$ is polynomial or logarithmic.

Empirically, the SW method is found to mix rapidly. Recently some analytic results on its performance have surfaced. Cooper and Frieze (1999) proved using a path coupling technique that SW mixs rapidly on sparsely connected graphs.

Theorem 1 (Cooper and Frieze 1999) *Let $n = |V|$ and Δ be the maximum number of edges at any single vertex, and* L *the number of colors in Potts model. If* $\mathbf{G}$ *is a tree, then the SW mixing time is $O(n)$ for any β and* L. *If $\Delta = O(1)$, then there exists $\rho_o = \rho(\Delta)$ such that if $\rho \leq \rho_o$ (i.e. higher than a certain temperature), then SW has polynomial mixing time for all* L.

A negative case was constructed by Gore and Jerrum (1997) on complete graph.

Theorem 2 (Gore and Jerrum 1997) *If* $\mathbf{G}$ *is a complete graph and* $\mathrm{L} > 2$, *then for* $\beta = \frac{2(\mathrm{L}-1)\ln(\mathrm{L}-1)}{n(\mathrm{L}-2)}$, *the SW does not mix rapidly.*

In the image analysis applications, our graph often observes the Copper-Frieze condition and the graph is far from being complete.

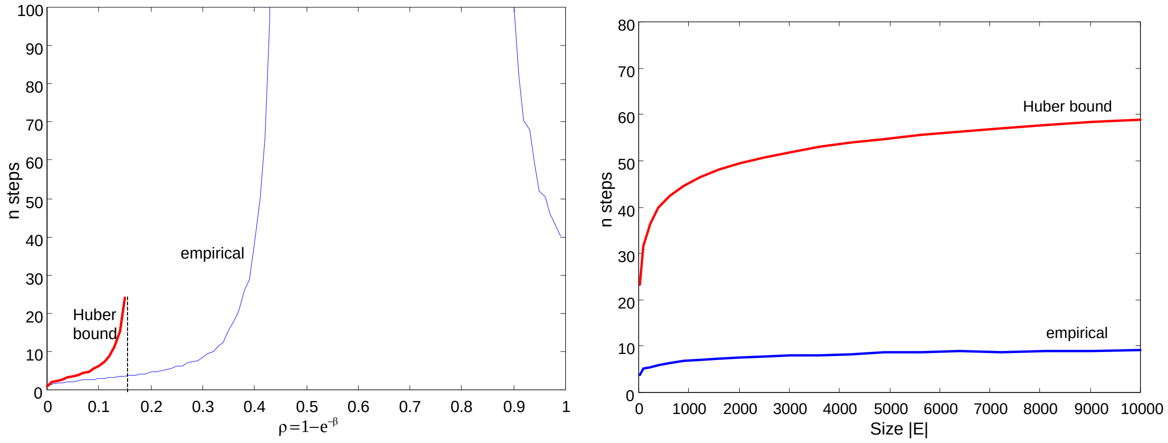
Most recently an exact sampling technique was developed for SW on Potts by Huber (2002) for very high or very low temperatures. It designs a bounding chain which assumes that each vertex $v_i \in V$ has a set of colors S_i initialized with the full set $|S_i| = \mathrm{L}$, $\forall i$. The Bernoulli probability for the auxiliary variables μ_{ij} is changed to

$$\mathbf{U}^{\mathrm{bd}} = \{\mu_{ij}^{\mathrm{bd}} : \mu_{ij}^{\mathrm{bd}} \in \{0,1\},\ \mu_{ij} \sim \mathrm{Bernoulli}(\rho\mathbf{1}(S_i \cap S_j \neq \emptyset))\}. \quad (18)$$

Thus $\mathbf{U}^{\mathrm{bd}}$ has more edges than $\mathbf{U}$ in the original SW chain, i.e. $\mathbf{U} \subset \mathbf{U}^{\mathrm{bd}}$. When $\mathbf{U}^{\mathrm{bd}}$ collapses to $\mathbf{U}$, then all SW chains starting with arbitrary initial states have collapsed into the current single chain. Thus it must have converged (exact sampling). The step for collapsing is called the "coupling time".

Theorem 3 (Huber 2002) *Let $n = |V|$ and $m = |E|$, at high temperature, $\rho < \frac{1}{2(\Delta-1)}$, the bounding chain couples completely by time $O(\ln(2m))$ with probability at least $1/2$. At lower temperature, $\rho \geq 1 - \frac{1}{m\mathrm{L}}$, then the coupling time is $O((m\mathrm{L})^2)$ with probability at least $1/2$.*

In fact the Huber bound is not very tight as one may expect. Fig. 2(a) plots the results on a 5 × 5 lattice with torus boundary condition on the Ising model for the empirical coupling time against $\rho = 1 - e^{-\beta}$. The coupling time is large near the critical temperature (didn't plot). The Huber bound for the high temperature starts with $\rho_o = 0.16$ and is plotted by the short curve. The bound for the low temperature starts with $\rho_o > 0.99$ which is not visible. Fig.2.(b) plots the coupling time at $\rho = 0.15$ against the graph size $m = |E|$ and the Huber bound.



(a) (b)

Figure 2: The coupling time empirical plots and the Huber bounds for Ising model.

Despite the encouraging success discussed above, the SW method is limited in two aspects.

Limit 1. It is only valid for the Ising and Potts models, and furthermore it requires that the number of colorings L is known. In many applications, such as image analysis, L is the number of objects (or image regions) which has to be inferred from the input data.

Limit 2. It slows down quickly in the presence of external field, i.e input data. For example, in the image analysis problem, our goal is to infer the label $\mathbf{X}$ from the input image $\mathbf{I}$ and the target probability is a Bayesian posterior probability where $\pi_{\mathrm{PTS}}(\mathbf{X})$ is used as a prior model,

$$\pi(\mathbf{X}) = \pi(\mathbf{X}|\mathbf{I}) \propto \mathcal{L}(\mathbf{I}|\mathbf{X})\pi_{\mathrm{PTS}}(\mathbf{X}). \tag{19}$$

$\mathcal{L}(\mathbf{I}|\mathbf{X})$ is the likelihood model, such as independent Gaussians $N(\bar{\mathbf{I}}_c, \sigma_c^2)$ for each coloring $c = 1, 2, ..., \mathrm{L}$,

$$\mathcal{L}(\mathbf{I}|\mathbf{X}) \propto \prod_{c=1}^{\mathrm{L}} \prod_{x_i=c} \frac{1}{\sqrt{2\pi}\sigma_c} \exp\{-\frac{(\mathbf{I}(v_i) - \bar{\mathbf{I}}_c)^2}{2\sigma_c^2}\}. \tag{20}$$

The slowing down is partially attributed to the fact that the Bernoulli probability $\rho = 1 - e^{-\beta}$ for the auxiliary variable is calculated independently of the input image.

2.4 SW Interpretation 2: slice sampling and decoupling

In the presence of external field (data), the SW method can be interpreted and extended by the auxiliary method proposed by Higdon (1998). Suppose we write the target probability in a more general form,

$$\pi(\mathbf{X}) = \frac{1}{Z} \prod_{v_i \in V} \phi_i(x_i) \cdot \prod_{<i,j> \in E} \psi(x_i, x_j), \quad \phi() > 0, \psi() > 0. \tag{21}$$

For the Potts model above, we have $\psi(x_i, x_j) = e^{\beta \mathbf{1}(x_i = x_j)}$. Higdon (1998) introduced a continuous variable on the edges as the *bond strength*,

$$W = \{\omega_{ij} : \ \omega_{ij} \in [0, +\infty), \forall < i, j > \in E\} \tag{22}$$

In contrast to the Bernoulli probability for the binary variable μ_{ij} in eqn. (3), the bond variables follow uniform probabilities, depending on $\mathbf{X}$,

$$\omega_{ij} | (x_i, x_j) \sim \text{Unif}[0, \psi(x_i, x_j)] = \psi^{-1}(x_i, x_j) \mathbf{1}(0 \leq \omega_{ij} \leq \psi(x_i, x_j)). \tag{23}$$

Thus a conditional probability is constructed as

$$p_{\text{HGD}}(W|\mathbf{X}) = \prod_{<i,j> \in E} p(\omega_{ij} | x_i, x_j) = \prod_{<i,j> \in E} \psi^{-1}(x_i, x_j) \mathbf{1}(0 \leq \omega_{ij} \leq \psi(x_i, x_j)). \tag{24}$$

This formula is chosen to cancel the internal field in a joint probability,

$$p_{\text{HGD}}(\mathbf{X}, W) = \pi(\mathbf{X}) p(W|\mathbf{X}) = \frac{1}{Z} [\prod_{v_i \in V} \phi_i(x_i)] \cdot [\prod_{<i,j> \in E} \mathbf{1}(0 \leq \omega_{ij} \leq \psi(x_i, x_j))]. \tag{25}$$

We have the second conditional probability by the Bayes rule,

$$p_{\text{HGD}}(\mathbf{X}|W) = \frac{1}{Z'} [\prod_{v_i \in V} \phi_i(x_i)] \cdot [\prod_{<i,j> \in E} \mathbf{1}(0 \leq \omega_{ij} \leq \psi(x_i, x_j))] \tag{26}$$

That is, given the bond strength ω_{ij}, x_i and x_j must achieve higher probability factor so that the condition $\psi(x_i, x_j) \geq \omega_{ij}$ is observed. This idea is called "slice sampling". In case of the Potts model, this becomes,

$$p(\mathbf{X}|W) = \frac{1}{Z'} [\prod_{v_i \in V} \phi_i(x_i)] \cdot [\prod_{<i,j> \in E} \mathbf{1}(0 \leq \omega_{ij} \leq e^{\beta \mathbf{1}(x_i = x_j)}] \tag{27}$$

Given W, the second product imposes a hard constraint on $\mathbf{X}$. If $\omega_{ij} \leq 1$, $\mathbf{1}(0 \leq \omega_{ij} \leq e^{\beta \mathbf{1}(x_i = x_j)}) = 1$ is satisfied for any x_i, x_j, because $\beta > 0$ and $e^{\beta \mathbf{1}(x_i = x_j)} \geq 1$. Thus it imposes no constraints on x_i, x_j. If $\omega_{ij} > 1$, then it imposes the constraint that $x_i = x_j$. Thus the auxiliary variables μ_{ij} and ω_{ij} are linked by the following equation,

$$\mu_{ij} = \mathbf{1}(\omega_{ij} > 1), \;\; \forall < i, j > \in E. \tag{28}$$

Thus we have to turn on the edges if $\omega_{ij} > 1$, otherwise we turn it off.

$$E_{\text{on}}(W) = \{e =< ij >: \;\; \omega_{ij} > 1, < i, j > \in E\}. \tag{29}$$

Given W, we have the set of connected components and the vertices in each component receive the same color.

$$\text{CP}(W) = \{\text{cp}_k : k = 1, 2, ..., K, \cup_{i=1}^{K} \text{cp}_k = V\}. \tag{30}$$

As the hard constraints are absorbed by the connected component, the conditional probability in eqn. (27) becomes

$$p_{\text{HGD}}(\mathbf{X}|W) = \prod_{k=1}^{K} \prod_{v_i \in \text{cp}_k} \phi_i(x_i). \tag{31}$$

As we can see, the coloring of each connected component is independent of other vertices (completely decoupled !). In the special case when $\phi_i(x_i) = 1$, it reduces to the RCM model in the previous subsection.

In summary $p_{\text{HGD}}(\mathbf{X}, W)$, like $p_{\text{ES}}(\mathbf{X}, \mathbf{U})$ in eqn.(7), has marginal probability being the target $\pi(\mathbf{X})$ and has two conditional probabilities that are easy to sample. There are two problems with this design.

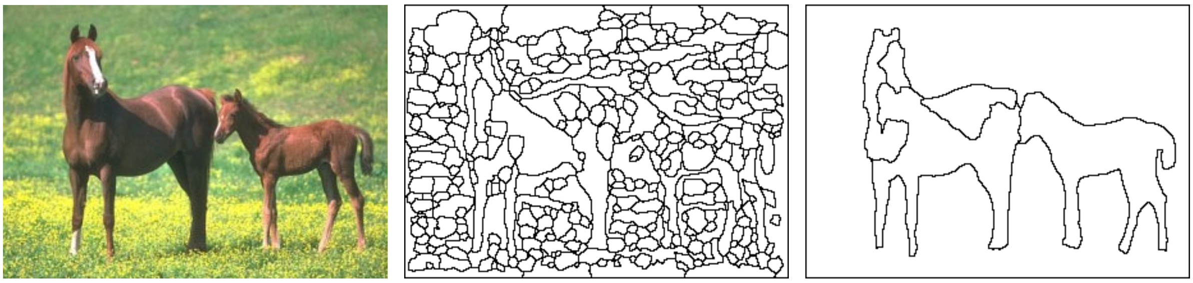
Firstly, although the decoupling idea with conditional probability $p_{\text{HGD}}(W|\mathbf{X})$ in eqn. (26) is valid for any pair clique Markov random field models and thus goes beyond the Potts model, the hard constraints may become impractical to compute for non-Potts model. That is, given W, the constraint conditions on $\mathbf{X}$ are no longer expressed as clustering. Many slice sampling methods suffer from this problem.

Secondly, although the flipping step in eqn.(31) makes use of the data, the clustering step in eqn. (24) does not. It is similar to the original SW method. This in practice often make the formed cluster ineffective.

3 Generalizing SW to arbitrary probabilities on graph

In this section, we generalize the SW to arbitrary probabilities from the perspective of Metropolis-Hastings method (Metropolis et al 1953, Hastings 1970). Our method iterates three steps: (i) a clustering step driven by data, (ii) a label flipping step which can introduce new labels, and (iii) an acceptance step for the proposed labelling. A key observation is the simple formula in calculating the acceptance probabilities.

We deliberate the three steps in the following three subsections, and then we show how it reduces to the original SW with Potts models.



(a). Input image (b). atomic regions (c). segmentation

Figure 3: Example of image segmentation. (a). Input image. (b). Atomic regions by edge detection followed by edge tracing and contour closing. each atomic region is a vertex in the graph **G**. c. Segmentation (labeling) result where each closed region is assigned a color or label.

We illustrate the algorithm by an example on image segmentation shown in Fig. 3. Fig. 3.(a) is an input image **I** on a lattice Λ, which is decomposed into a number of ”atomic regions” to reduce the graph size in a preprocessing stage. Each atomic region has nearly constant intensity and is a vertex in the graph **G**. Two vertices are connected if their atomic regions are adjacent (i.e. sharing boundary). Fig. 3.(c) is a result by our algorithm optimizing a Bayesian probability $\pi(\mathbf{X}) = \pi(\mathbf{X}|\mathbf{I})$ (see section (5) for details). The result **X** assigns a uniform color to all vertices in each close region which hopefully corresponds to an object in the scene or a part of it. Note that the number of objects or colors L is unknown, and we do not distinguish the different permutations of the labels.

3.1 Step 1: data-driven clustering

We augment the adjacency graph $\mathbf{G}$ with a set of binary variables on the edges $\mathbf{U} = \{\mu_{ij} :< i,j >\in E\}$, as in the original SW method. Each μ_{ij} follows a Bernoulli probability depending on the current state of the two vertices x_i and x_j,

$$\mu_{ij}|(x_i,x_j) \;\sim\; \text{Bernoulli}(q_{ij}\mathbf{1}(x_i = x_j)),\;\; \forall < i,j >\in E. \tag{32}$$

q_{ij} is a probability on edge $< i,j >$ which tells how likely the two vertices v_i and v_j have the same label. In Bayesian inference where the target $\pi(\mathbf{X})$ is a posterior probability, then q_{ij} can be better informed by the data.

For the image segmentation example, q_{ij} is computed based on the similarity between image intensities at v_i and v_j (or their local neighborhood) and it may be an approximate to the marginal probability of $\pi(\mathbf{X}|\mathbf{I})$,

$$q_{ij} = q(x_i = x_j|\mathbf{I}(v_i),\mathbf{I}(v_j)) \;\;\approx\;\; \pi(x_i = x_j|\mathbf{I}). \tag{33}$$

There are many ways for computing $q(x_i = x_j|\mathbf{I}(v_i),\mathbf{I}(v_j))$ using so called discriminative methods, and it is beyond this paper to discuss details.

Our method will work for any q_{ij}, but a good approximation will inform the clustering step and achieve faster convergence empirically. Fig. 4 shows nine clustering examples of the horse image. In these examples, we set all vertices to the same color ($\mathbf{X} = c$) and sample the edge probability independently,

$$\mathbf{U}|\mathbf{X} = c \;\sim\; \prod_{<i,j>\in E} \text{Bernoulli}(q_{ij}). \tag{34}$$

The connected components in CP($\mathbf{U}$) are shown by different regions. We repeat the clustering step nine times. As we can see, the edge probabilities lead to ”meaningful” clusters which correspond to distinct objects in the image. Such effects cannot be observed using constant edge probability.

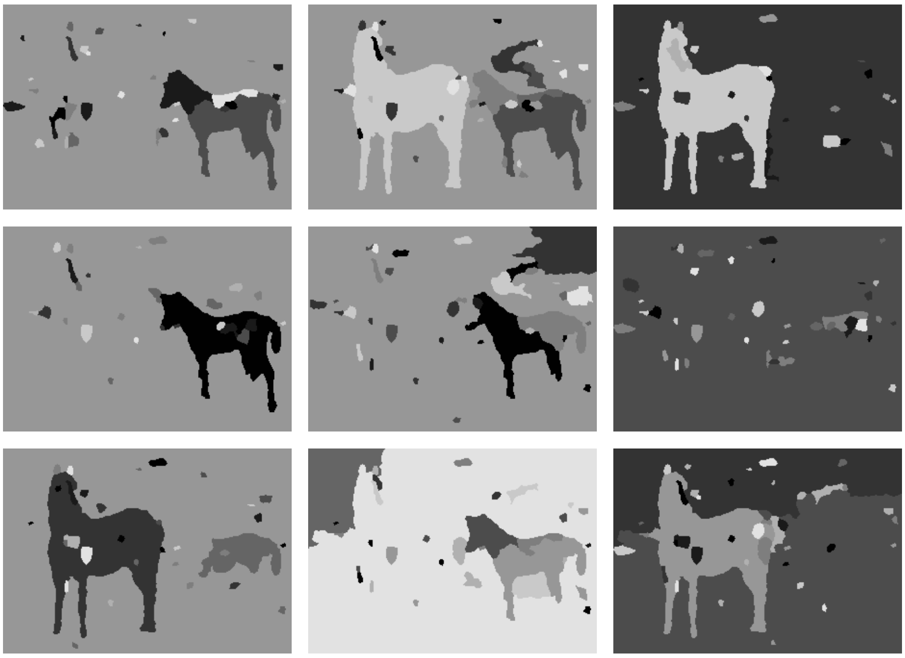


Figure 4: Nine examples of the connected components for the horse image computed using discriminative edge probabilities given that $\mathbf{X}$ is a uniform color $\mathbf{X} = c$ for all vertices.

3.2 Step 2: flipping of color

Let $\mathbf{X} = (V_1, V_2, ..., V_n)$ be the current coloring state, and the edge variables $\mathbf{U}$ sampled conditional on $\mathbf{X}$ further decompose $\mathbf{X}$ into a number of connected components

$$\mathrm{CP}(\mathbf{U}|\mathbf{X}) = \{\mathrm{cp}_i : i = 1, 2, ..., N(\mathbf{U}|\mathbf{X})\}. \tag{35}$$

Suppose we select one connected component $R \in \mathrm{CP}(\mathbf{U}|\mathbf{X})$ with color $\mathbf{X}_R = \ell \in \{1, 2, ..., n\}$, and assign its color to $\ell' \in \{1, 2, ..., n, n+1\}$ with probability $q(l'|R, \mathbf{X})$ (to be designed shortly), obtaining new state $\mathbf{X}'$. There are three cases shown in Fig. 5.

1. The canonical case: $R \subset V_\ell$ and $\ell' \leq n$. That is, a portion of V_ℓ is re-grouped into an existing color $V_{\ell'}$, and the number of colors remains $\mathrm{L} = n$ in π'. The moves between $\mathbf{X}_A \leftrightarrow \mathbf{X}_B$ in Fig. 5 are examples.

2. The merge case: $R = V_\ell$ in $\mathbf{X}$ is the set of all vertices that have color ℓ and $\ell' \leq n$, $\ell \neq \ell'$. That is, color V_ℓ is merged to $V_{\ell'}$, and the number of distinct colors reduces to $n - 1$ in $\mathbf{X}'$. The moves $\mathbf{X}_C \to \mathbf{X}_A$ or $\mathbf{X}_C \to \mathbf{X}_B$ in Fig. 5 are examples.

3. The split case: $R \subset V_\ell$ and $\ell' = n+1$. V_ℓ is split into two pieces and the number of distinct color increases to $n+1$ in $\mathbf{X}'$. The moves $\mathbf{X}_A \to \mathbf{X}_C$ in Fig.5 are examples.

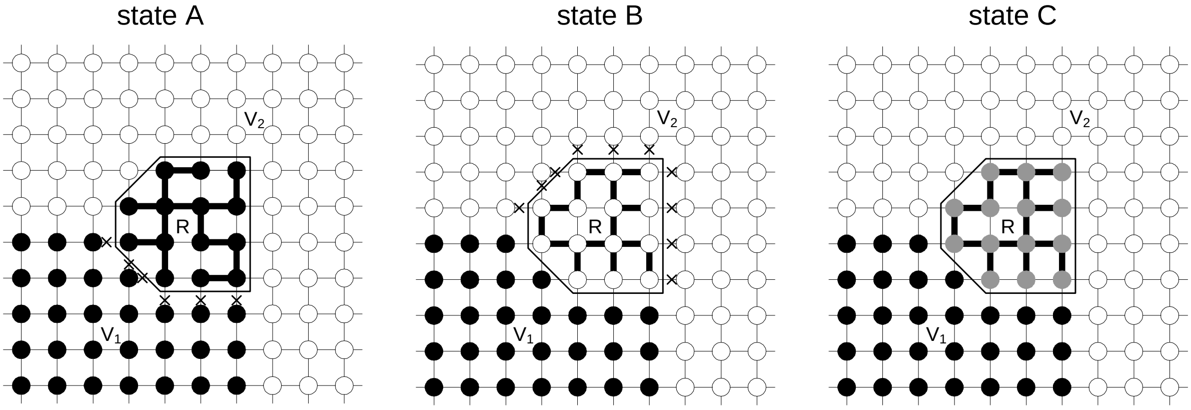


Figure 5: Three labeling states $\mathbf{X}_A, \mathbf{X}_B, \mathbf{X}_C$ which differ only in the color of a cluster R.

Note that this color flipping step is also different from the original SW with Potts model as we allow new colors in each step. The number of color L is not fixed.

3.3 Step 3: accepting the flipping

The previous two steps basically have proposed a move between two states $\mathbf{X}$ and $\mathbf{X}'$ which differ in coloring a connected component R. In the third step we accept the move with probability,

$$\alpha(\mathbf{X} \to \mathbf{X}') = \min\{1, \frac{q(\mathbf{X}' \to \mathbf{X})}{q(\mathbf{X} \to \mathbf{X}')} \cdot \frac{\pi(\mathbf{X}')}{\pi(\mathbf{X})}\}. \tag{36}$$

$q(\mathbf{X}' \to \mathbf{X})$ and $q(\mathbf{X} \to \mathbf{X}')$ are the proposal probabilities between $\mathbf{X}$ and $\mathbf{X}'$. If the proposal is rejected, the Markov chain stays at state $\mathbf{X}$. The transition kernel is

$$\mathcal{K}(\mathbf{X} \to \mathbf{X}') = q(\mathbf{X} \to \mathbf{X}')\alpha(\mathbf{X} \to \mathbf{X}'), \quad \forall \mathbf{X} \neq \mathbf{X}'. \tag{37}$$

For the canonical case, there is a unique path for moving between bX and $\mathbf{X}'$ in one step – choosing R and changing its color. The proposal probability ratio is the product of two ratios decided by the clustering and flipping steps respectively: (i) the probability ratio for selecting R as the candidate in the clustering step in both states $\mathbf{X}$ and $\mathbf{X}'$, and (ii) the probability ratio for selecting the new labels for R in the flipping step.

$$\frac{q(\mathbf{X}' \to \mathbf{X})}{q(\mathbf{X} \to \mathbf{X}')} = \frac{q(R|\mathbf{X}')}{q(R|\mathbf{X})} \cdot \frac{q(\mathbf{X}_R = \ell|R, \mathbf{X}')}{q(\mathbf{X}_R = \ell'|R, \mathbf{X})}. \tag{38}$$

For the split and merge cases, there are two paths between $\mathbf{X}$ and $\mathbf{X}'$. But this does not change the conclusion (see Appendix B).

Now we compute the probability ratio $\frac{q(R|\mathbf{X}')}{q(R|\mathbf{X})}$ for proposing R.

Definition 1 *Let $\mathbf{X} = (V_1, V_2, ..., V_{\rm L})$ be a coloring state, and $R \in {\rm CP}(U|\mathbf{X})$ a connected component, the "cut" between R and V_k is a set of edges between R and $V_k \backslash R$,*

$$\mathcal{C}(R, V_k) = \{< i, j >: \; i \in R, j \in V_k \backslash R\}, \;\; \forall k.$$

One of our key observation is that this ratio only depends on the cuts between R and rest vertices.

Proposition 3 *In the above notation, we have*

$$\frac{q(R|\mathbf{X})}{q(R|\mathbf{X}')} = \frac{\prod_{<i,j> \in \mathcal{C}(R,V_\ell)}(1 - q_{ij})}{\prod_{<i,j> \in \mathcal{C}(R,V_{\ell'})}(1 - q_{ij})}. \tag{39}$$

q_{ij}'s are the edge probabilities.

[Proof] We put the proof in the appendix A for clarity.

The crosses in Fig.5.(a) and (b) show the cut $\mathcal{C}(R, V_1)$ and $\mathcal{C}(R, V_2)$ respectively. In Fig.5.(c), $R = V_3$ and thus $\mathcal{C}(R, V_3) = \emptyset$ and $\prod_{<i,j> \in \mathcal{C}(R,V_3)}(1 - q_{ij}) = 1$.

Summarizing the results in eqns.(36), (38) and (39), we have the following theorem.

Theorem 4 *The acceptance probability for the proposed cluster flipping is,*

$$\alpha(\mathbf{X} \to \mathbf{X}') = \min\{1, \frac{\prod_{<i,j> \in \mathcal{C}(R,V_{\ell'})}(1 - q_{ij})}{\prod_{<i,j> \in \mathcal{C}(R,V_\ell)}(1 - q_{ij})} \cdot \frac{q(\mathbf{X}_R = \ell|R, \mathbf{X}')}{q(\mathbf{X}_R = \ell'|R, \mathbf{X})} \cdot \frac{\pi(\mathbf{X}')}{\pi(\mathbf{X})}\}. \tag{40}$$

[Proof] The proof is given in Appendix B. It has to account for the split and merge cases which have two possible paths between the states $\mathbf{X}$ and $\mathbf{X}'$.

Example. In image analysis, $\pi(\mathbf{X})$ is a Bayesian posterior $\pi(\mathbf{X}|\mathbf{I}) \propto \mathcal{L}(\mathbf{I}|\mathbf{X})p_o(\mathbf{X})$ with the prior probability $p_o(\mathbf{X})$ being a Markov model (like Potts in Eqn. (20)). Therefore one can compute the ratio of the target probabilities in the local neighborhood of R, i.e. ∂R.

$$\frac{\pi(\mathbf{X}')}{\pi(\mathbf{X})} = \frac{\mathcal{L}(\mathbf{I}_R|\mathbf{X}_R = \ell')}{\mathcal{L}(\mathbf{I}_R|\mathbf{X}_R = \ell)} \cdot \frac{p_o(\mathbf{X}_R = \ell'|\mathbf{X}_{\partial R})}{p_o(\mathbf{X}_R = \ell|\mathbf{X}_{\partial R})} \tag{41}$$

Note that $\mathbf{X}_{\partial R} = \mathbf{X}'_{\partial R}$ in the above equation.

The second ratio in eq. (40) is easy to design. For example, we can make it proportional to the likelihood,

$$q(\mathbf{X}_R = \ell | R, \mathbf{X}) = \mathcal{L}(\mathbf{I}_R | \mathbf{X}_R = \ell), \quad \forall \ell. \tag{42}$$

Therefore,

$$\frac{q(\mathbf{X}_R = \ell | R, \mathbf{X}')}{q(\mathbf{X}_R = \ell' | R, \mathbf{X})} = \frac{\mathcal{L}(\mathbf{I}_R | \mathbf{X}_R = \ell)}{\mathcal{L}(\mathbf{I}_R | \mathbf{X}_R = \ell')} \tag{43}$$

It cancels the likelihood ratio in eqn.(41). We get

Proposition 4 *The acceptance probability for the proposed cluster flipping using the proposal (42) is,*

$$\alpha(\mathbf{X} \to \mathbf{X}') = \min\{1, \frac{\prod_{<i,j> \in \mathcal{C}(R, V_{\ell'})}(1 - q_{ij})}{\prod_{<i,j> \in \mathcal{C}(R, V_\ell)}(1 - q_{ij})} \cdot \frac{p_o(\mathbf{X}_R = \ell' | \mathbf{X}_{\partial R})}{p_o(\mathbf{X}_R = \ell | \mathbf{X}_{\partial R})}\}. \tag{44}$$

The result above states that the computation is limited to a local neighborhood of R defined by the prior model. This result is also true if one changes the clustering step by growing R from a vertex, i.e. the Wolff modification.

In the experiments on image analysis, our cluster sampling method is empirically $O(100)$ times faster than the single site Gibbs sampler in terms of CPU time. We refer to plots and comparison in Figs.(8), (9) and (10) in section (5) for details.

3.4 SW Interpretation 3: the Metropolis-Hastings perspective

Now we are ready to derive the original SW method as a special case.

Proposition 5 *If we set the edge probability to a constant $q_{ij} = 1 - e^{-\beta}$, then*

$$\frac{q(R|\mathbf{X})}{q(R|\mathbf{X}')} = \frac{\prod_{<i,j> \in \mathcal{C}(R, V_\ell)}(1 - q_{ij})}{\prod_{<i,j> \in \mathcal{C}(R, V_{\ell'})}(1 - q_{ij})} = \exp\{\beta(|\mathcal{C}(R, V_{\ell'})| - |\mathcal{C}(R, V_\ell)|)\}, \tag{45}$$

where $|\mathcal{C}|$ is the cardinality of the set.

As $\mathbf{X}$ and $\mathbf{X}'$ only differ in labeling R, the potentials for the Potts model only differs at the "cracks" between R and V_ℓ and $V_{\ell'}$ respectively.

Proposition 6 *For the Potts model $\pi(\mathbf{X}) = p_o(\mathbf{X}) = \pi_{\mathrm{PTS}}(\mathbf{X})$,*

$$\frac{\pi_{\mathrm{PTS}}(\mathbf{X}_R = \ell' | \mathbf{X}_{\partial R})}{\pi_{\mathrm{PTS}}(\mathbf{X}_R = \ell | \mathbf{X}_{\partial R})} = \exp\{\beta(|\mathcal{C}(R, V_\ell)| - |\mathcal{C}(R, V_{\ell'})|)\} \tag{46}$$

Therefore, following eq. (40) (where the proposal probabilities for the labels are uniform), the acceptance probability for the Potts model is always one, due to cancellation.

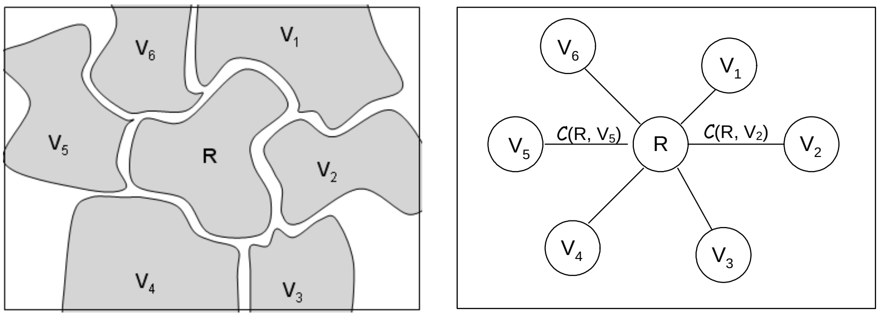
$$\alpha(\mathbf{X} \to \mathbf{X}') = 1. \tag{47}$$

Therefore the third acceptance step is always omitted. This interpretation is related to the Wolff (1989) modification (see also Liu 2001, p157).

4 Variants of the cluster sampling method

In this section, we briefly discuss two variants of the cluster sampling method.

4.1 Cluster Gibbs sampling — the "hit-and-run" perspective



(a) (b)

Figure 6: Illustrating the cluster Gibbs sampler. (a) The cluster R has a number of neighboring components of uniform color. (b) The cuts between R and its neighboring colors. The sampler follows a conditional probability modified by the edge strength defined on the cuts.

With a slight change, we can modify the cluster sampling method to a generalized Gibbs sampler.

Suppose that $R \in \mathrm{CP}(U|\mathbf{X})$ is the candidate chosen in the clustering step, and Fig. 6 shows its cuts with adjacent sets

$$\mathcal{C}(R, V_k),\ k = 1, 2, ..., \mathrm{L}(\mathbf{X}).$$

We compute the edge weight γ_k as the strength of connectivity between R and $V_k \backslash R$,

$$\gamma_k = \prod_{<i,j> \in \mathcal{C}(R, V_k)} (1 - q_{ij}). \tag{48}$$

Proposition 7 *Let $\pi(\mathbf{X})$ be the target probability, in the notation above. If R is relabelled probabilistically with*

$$q(\mathbf{X}_R = k | R, \mathbf{X}) \propto \gamma_k \pi(\mathbf{X}_R = k | \mathbf{X}_{\partial R}), \quad k = 1, 2, ..., N(\mathbf{X}), \tag{49}$$

then the acceptance probability is always 1 in the third step.

[Proof] See Appendix C.

This yields a generalized Gibbs sampler which flips the color of a cluster according to a modified conditional probability.

Cluster Gibbs Sampler

1. Cluster step: choosing a vertex $v \in V$ and group a cluster R from v by the Bernoulli edge probability μ_{ij}.

2. Flipping step: relabel R according to eqn. (49).

The tranditional single site Gibbs sampler (Geman and Geman 1984) is a special case when $q_{ij} = 0$ for all $< i, j >$ and thus $R = \{v\}$ and $\gamma_k = 1$ for all k.

One may also view the above method from the perspective of hit-and-run. In continuous state space, a hit-and-run method (see Gilks etc 1996) chooses a new direction $\vec{e}$ (random ray) at time t and then sample on this direction by $a \sim \pi(x + a\vec{e})$. Liu and Wu (1999) extended it ray to any compact groups of actions. In finite state space Ω, one can choose any finite sets $\Omega_a \subset \Omega$ and then apply the Gibbs sampler within the set. [1]

But it is difficult to choose good directions or subsets in hit-and-run methods. In the cluster Gibbs sampler presented above, the subset is selected by the auxiliary variables on the edges.

[1] Persi Diaconis once discussed a unifying view of hit-and-run for MCMC in a talk in 2002.

4.2 The multiple flipping scheme

Given a set of connected components $\mathrm{CP}(\mathbf{U}|\mathbf{X})$ (see eqn. (35)) after the clustering step, instead of flipping a single component R, we can flip all (or any chosen number of) connected components simultaneously. There is room for designing the proposal probabilities for labeling these connected components, independently or jointly. In what follows, we assume the labels are chosen independently for each connected component $\mathrm{cp} \in \mathrm{CP}(\mathbf{U}|\mathbf{X})$, by sampling from a proposal probability $q(\mathbf{X}_{\mathrm{cp}} = l|\mathrm{cp})$. Suppose we obtain a new label $\mathbf{X}'$ after flipping. Let $E_{\mathrm{on}}(\mathbf{X}) \subset E$ and $E_{\mathrm{on}}(\mathbf{X}') \subset E$ be the subsets of edges that connect the vertices of same color in $\mathbf{X}$ and $\mathbf{X}'$ respectively. We define two cuts by the differences of the sets

$$\mathcal{C}(\mathbf{X} \to \mathbf{X}') = E_{\mathrm{on}}(\mathbf{X}') - E_{\mathrm{on}}(\mathbf{X}), \text{ and } \mathcal{C}(\mathbf{X}' \to \mathbf{X}) = E_{\mathrm{on}}(\mathbf{X}) - E_{\mathrm{on}}(\mathbf{X}'), \tag{50}$$

We denote the set of connected components which have different colors before and after the flipping by $D(\mathbf{X}, \mathbf{X}') = \{\mathrm{cp} : \mathbf{X}_{\mathrm{cp}} \neq \mathbf{X}'_{\mathrm{cp}}\}$.

Proposition 8 *The acceptance probability of the multiple flipping scheme is*

$$\alpha(\mathbf{X} \to \mathbf{X}') = \min\{1, \frac{\prod_{<i,j>\in\mathcal{C}(\mathbf{X}\to\mathbf{X}')}(1-q_{ij})}{\prod_{<i,j>\in\mathcal{C}(\mathbf{X}'\to\mathbf{X})}(1-q_{ij})} \frac{\prod_{\mathrm{cp}\in D(\mathbf{X},\mathbf{X}')} q(\mathbf{X}'_{\mathrm{cp}}|\mathrm{cp})}{\prod_{\mathrm{cp}\in D(\mathbf{X},\mathbf{X}')} q(\mathbf{X}_{\mathrm{cp}}|\mathrm{cp})} \cdot \frac{p(\pi')}{p(\pi)}\} \tag{51}$$

[Proof] See Appendix D.

Observe that when $D = \{R\}$ is a single connected component, this reduces to Theorem 4.

It is worth mentioning that if we flip all connected components simultaneously, then the Markov transition graph of $\mathcal{K}(\mathbf{X}, \mathbf{X}')$ is fully connected, i.e.

$$\mathcal{K}(\mathbf{X}, \mathbf{X}') > 0, \quad \forall \mathbf{X}, \mathbf{X}' \in \Omega. \tag{52}$$

This means that the Markov chain can walk between any two partitions in a single step.

5 Experiment 1: image segmentation

Our first experiment tests the cluster sampling algorithm in an image segmentation task. The objective is to partition the image into a number of disjoint regions (as Figs.3 and

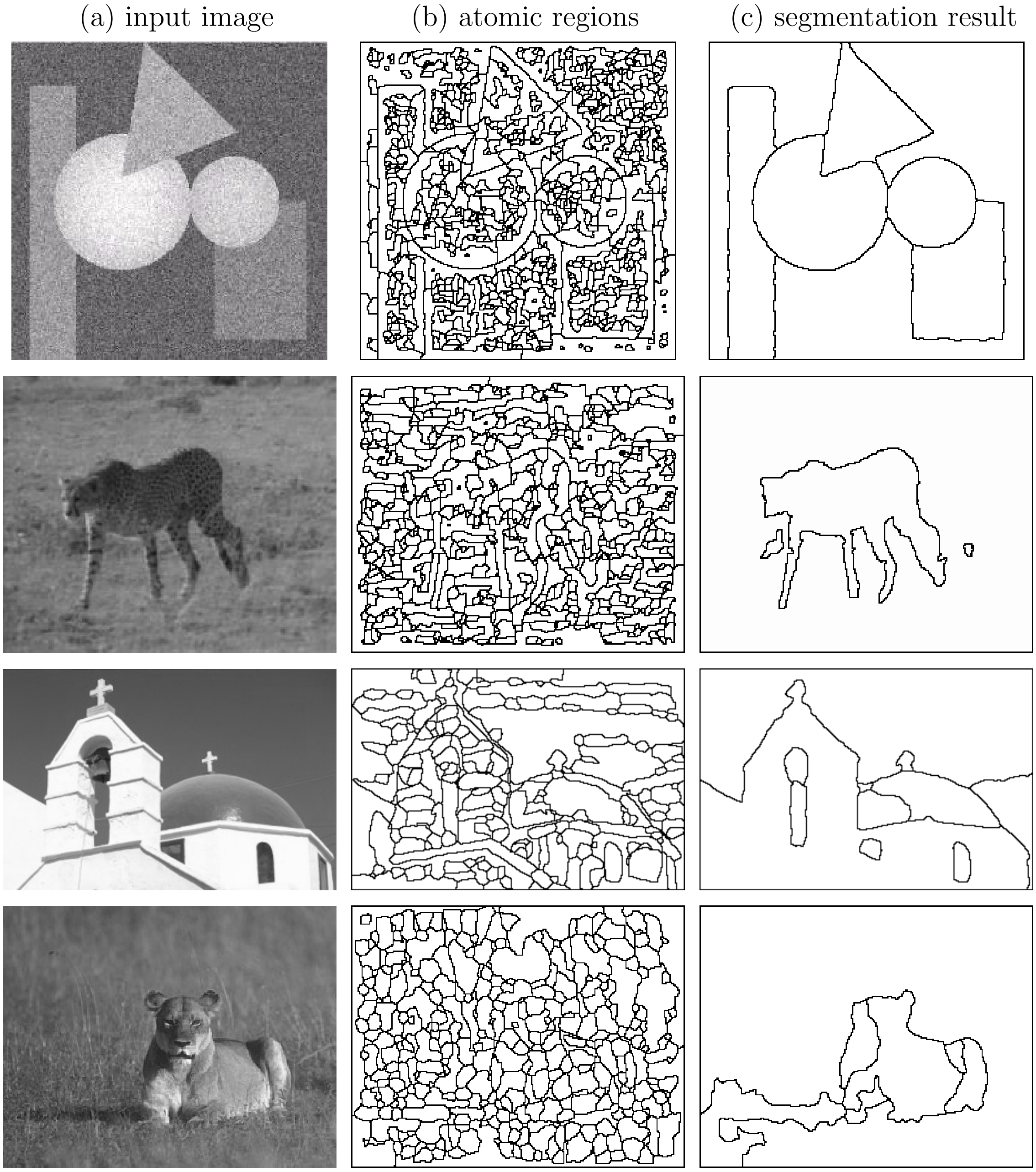


Figure 7: More results for image segmentation.

4 have shown) so that each region has consistent intensity in the sense of fitting to some image models. The final result should optimize a Bayesian posterior probability $\pi(\mathbf{X}) \propto \mathcal{L}(\mathbf{I}|\mathbf{X})p_o(\mathbf{X})$.

In such problem, $\mathbf{G}$ is an adjacency graph with vertices V being a set of atomic regions (see Figs.(3) and (4)). Usually $|V| = O(10^2)$. For each atomic region $v \in V$, we compute a 15-bin intensity histogram h normalized to 1. Thus the edge probability is calculated as

$$q_{ij} = p(\mu_e = \text{on}|\mathbf{I}(v_i), \mathbf{I}(v_j)) = \exp\{-\frac{1}{2}(KL(h_i||h_j) + KL(h_j||h_i))\}, \tag{53}$$

where $KL()$ is the Kullback-Leibler divergence between the two histograms. Usually q_{ij} should be close to zero for $< i, j >$ crossing object boundary. In our experiments, the edge probability leads to good clustering as Fig. 4 shows.

Now we briefly define the target probability in this experiment. Let $\mathbf{X} = (V_1, ..., V_{\mathrm{L}})$ be a coloring of the graph with L being a unknown variable, and the image intensities in each set V_k is consistent in terms of fitting to a model θ_k. Different colors are assumed to be independent. Therefore, we have,

$$\pi(\mathbf{X}) = \pi(\mathbf{X}|\mathbf{I}) \propto \prod_{k=1}^{\mathrm{L}} [\mathcal{L}(\mathbf{I}(V_k); \theta_k) p_o(\theta_k)] p_o(\mathbf{X}). \tag{54}$$

We selected three types of simple models for the likelihood models to account for different image properties. The first model is a non-parametric histogram $\mathcal{H}$, which in practice is represented by a vector of B-bins $(\mathcal{H}_1, ..., \mathcal{H}_B)$ normalized to 1. It accounts for cluttered objects, like vegetation.

$$\mathbf{I}(x, y; \theta_0) \sim \mathcal{H} \text{ iid}, \ \forall (x, y) \in V_k. \tag{55}$$

The other two are regression models for the smooth change of intensities in the two-dimensional image plane (x, y), and the residues follow the empirical distribution $\mathcal{H}$ (i.e. the histogram).

$$\mathbf{I}(x, y; \theta_1) = \beta_0 + \beta_1 x + \beta_2 y + \mathcal{H} \text{ iid}, \ \forall (x, y) \in V_k. \tag{56}$$

$$\mathbf{I}(x, y; \theta_2) = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 xy + \beta_5 y^2 + \mathcal{H} \text{ iid}, \ \forall (x, y) \in V_k. \tag{57}$$

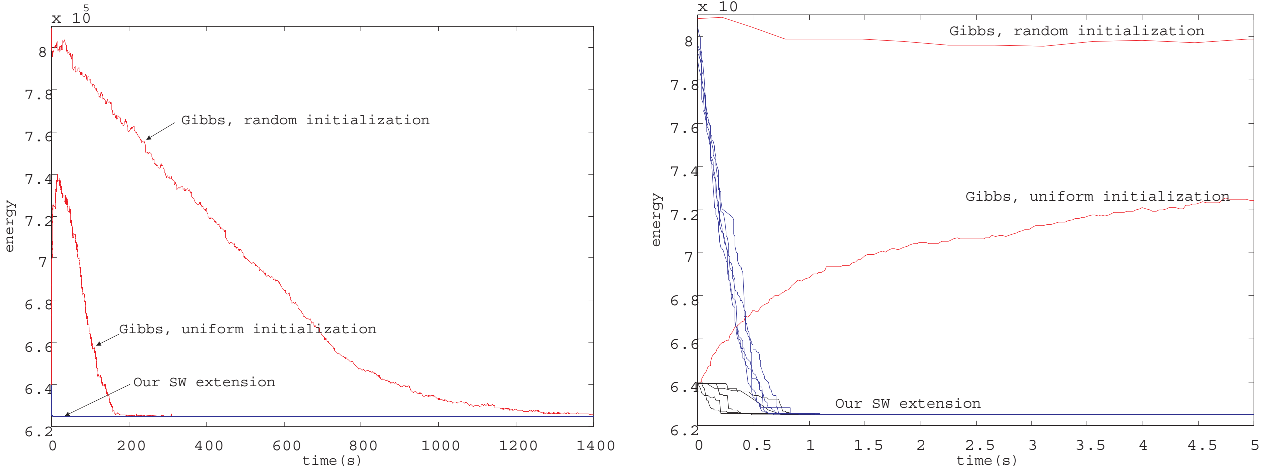
In all cases, the likelihood is expressed in terms of the entropy of the histogram $\mathcal{H}$

$$\mathcal{L}(\mathbf{I}(V_k); \theta_k) \propto \prod_{v \in V_k} \mathcal{H}(\mathbf{I}_v) = \prod_{j=1}^{B} \mathcal{H}_j^{n_j} = \exp(-|V_k| \text{entropy}(\mathcal{H})). \tag{58}$$

The model complexity is penalized by a prior probability $p_o(\theta_k)$ and the parameters θ in the above likelihoods are computed deterministically at each step as the best least square fit. The deterministic fitting could be replaced by the reversible jumps together with the flipping of color. This was done in (Tu and Zhu, 2002) and is beyond the scope of our experiments.

The prior model $p_o(\mathbf{X})$ encourages large and compact regions with small number of colors, as it was suggested in (Tu and Zhu 2002). Let $r_1, r_2, ..., r_m$, $m \geq \mathrm{L}$ be the connected components of all $V_k, k = 1, ..., \mathrm{L}$. Then the prior is

$$p_o(\mathbf{X}) \propto \exp\{-\alpha_0 \mathrm{L} - \alpha_1 m - \alpha_2 \sum_{k=1}^{m} \mathrm{Area}(r_k)^{0.9}\}. \tag{59}$$



(a) convergence CPU time in seconds

(b) Zoomed-in view of the first 5 seconds.

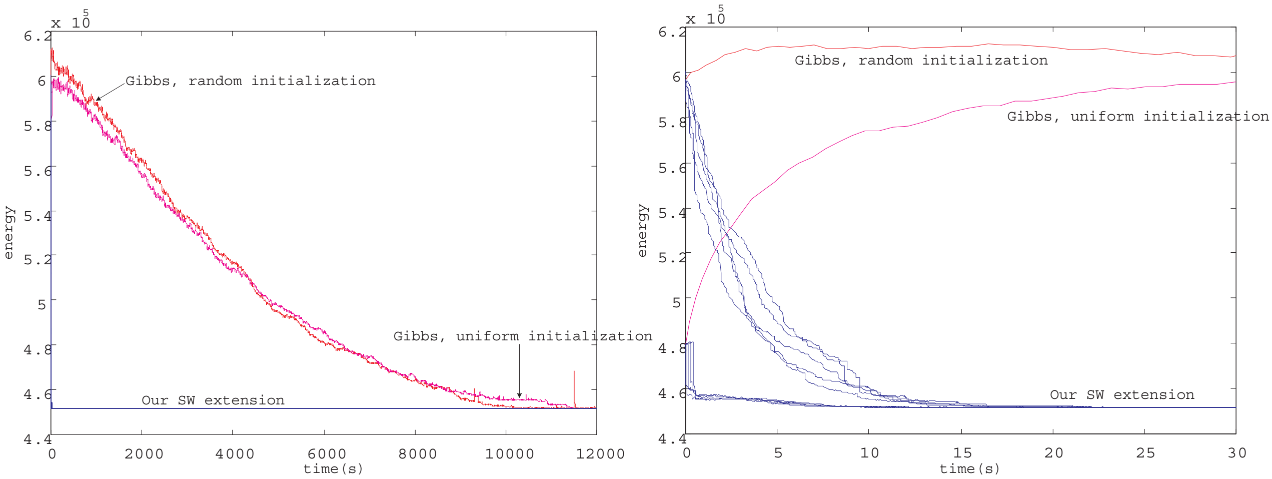
Figure 8: The plot of $-\ln \pi(X)$ over computing time for both the Gibbs sampler and our algorithm for the horse image. Both algorithms are measured by the CPU time in seconds using a Pentium IV PC. So they are comparable. (a). Plot in the first 1,400 seconds. The Gibbs sampler needs a high initial temperature and slow annealing step to achieve the same energy level. (b). The zoomed-in view of the first 5 seconds.

For the image segmentation example (horse) shown in Figs. 3 and 4, we compare the cluster sampling method with the single-site Gibbs sampler and the results are displayed in Fig. 8. Since our goal is to maximize the posterior probability $\pi(\mathbf{X})$, we must add an annealing scheme with a high initial temperature T_o and then decreases to a low temperature (0.05 in our experiments). We plot the $-\ln \pi(\mathbf{X})$ over CPU time in seconds with a Pentium IV PC. The Gibbs sampler needs to raise the initial temperature high (say $T_o \geq 100$)) and uses a slow annealing schedule to reach good solution. The cluster sampling method can run at low temperature. We usually raise the initial temperature to $T_o \leq 15$ and use a fast annealing scheme. Fig. 8.(a) plots the two algorithms at the first 1,400 seconds, and

Fig. 8.(b) is a zoomed-in view for the first 5 seconds.

We run the two algorithms with two initializations. One is a random labeling of the atomic regions and thus has higher $-\ln \pi(\mathbf{X})$, and the other initialization sets all vertices to the same color. The clustering methods are run five times on both cases. They all converged to one solution (see Fig.3.(c)) within 1 second, which is $O(10^2)$ times faster than the Gibbs sampler.

Fig.7 shows four more images. Using the sample comparison method as in the horse image, we plot $-\ln \pi(\mathbf{X})$ against running time in Figs. 9 and 10 for the images in the first and second row of Fig.7 respectively. In experiments, we also compared the effect of the edge probabilities. The clustering algorithm are $O(100)$ times slower if we use a constant edge probability $\mu_{ij} = c \in (0,1)$ as the original SW method does. For example the single-site Gibbs sampler is an example with $q_{ij} = 0$, $\forall\, i, j$.



(a) (b)

Figure 9: Convergence comparison between the clustering method and Gibbs sampler in CPU time (seconds) on the artificial image (circles, triangle and rectangles) in the first row of Fig.7. (a). The first 1,200 seconds. (Right) Zoomed-in view of the first 30 seconds. The clustering algorithm is run 5 trials for both the random and uniform initializations.

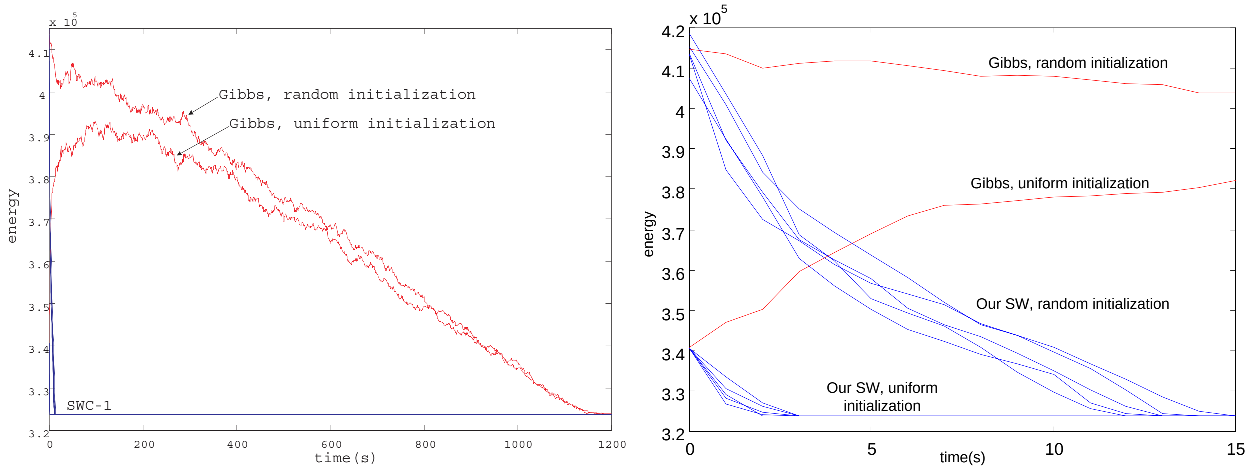


Figure 10: Convergence comparison between the clustering method and Gibbs sampler in CPU time (seconds) on the cheetah image. (Left) The first 1,200 seconds. (Right) Zoomed-in view of the first 15 seconds. The clustering algorithm is run 5 times for both the random and uniform initializations.

6 Multi-grid and Multi-level cluster sampling

When the graph size $\mathbf{G}$ is big, for example, $|V| = O(10^4) \sim O(10^6)$ in image analysis, a clustering step has to flip many edges and is costly computationally. This section presents two strategies for improving the speed – the multi-grid and multi-level cluster sampling. Our methods are different from the multi-grid and multi-level samplings ideas in the statistical literature (see Gilks et al 1996 and Liu 2001)

6.1 Rationale for multi-grid and multi-level cluster sampling

In multi-grid clustering sampling, we introduce an ”attention window” Λ (see Fig.12) which may change location and size over time. The cluster sampling is limited to within the window at each step, and this is equivalent to sampling a conditional probability,

$$\mathbf{X}_\Lambda \sim \pi(\mathbf{X}_\Lambda | \mathbf{X}_{\bar{\Lambda}}). \tag{60}$$

The multi-level cluster sampling is motivated by the problem of hierarchic graph labeling. Fig. 11 illustrates an example in motion segmentation. Suppose we are given two consecutive image frames in a video, and our goal consists of three parts: (i) calculate the

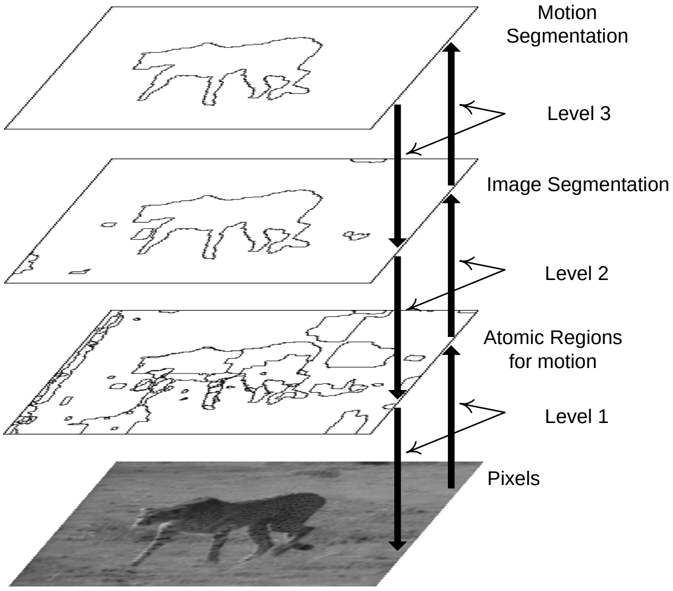


Figure 11: Cluster sampling on multi-level of graphs for motion segmentation. A connected component with the same color is frozen and collapsed into a single vertex in the level above.

planar velocity (i.e. optical flow) of the pixels in the second frame based on the displacement between pixels in two frames, (ii) segment (group) the pixels into regions of coherent intensities, and (iii) further group the regions into moving objects, such as the running cheetah and the grass background where each object should have both consistent intensity and motion velocity in the image planar.

This problem can be represented in a three-level labeling with $\mathbf{X} = (\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)})$, and this label forms three levels of graph shown in Fig. 11,

$$\{\mathbf{G}^{(s)} =< V^{(s)}, E^{(s)} >: \ s = 0, 1, 2\}. \tag{61}$$

$\mathbf{G}^{(0)}$ is the image lattice with each vertex being a pixel. The pixels are labeled by $\mathbf{X}^{(0)}$ according to their planar motion velocity and thus grouped into a number of small regions of nearly constant velocity in $\mathbf{G}^{(1)}$. The vertices in $\mathbf{G}^{(1)}$ are further labeled by $\mathbf{X}^{(1)}$ according to their intensities and grouped into a smaller graph $\mathbf{G}^{(2)}$, which is in turn labeled by $\mathbf{X}^{(2)}$. The vertices has reduced from $O(10^5)$ in $\mathbf{G}^{(0)}$ to $O(10^2)$ in $\mathbf{G}^{(1)}$ and to $O(10)$ in $\mathbf{G}^{(2)}$.

We should discuss more details in the next two subsections. In the rest of this subsection, we discuss the theoretical justifications for the multi-grid and multi-level cluster sampling.

The essence of the cluster sampling design is that its Markov chain kernel observes the detailed balance equations as a result of the Metropolis-Hastings design.

$$\pi(\mathbf{X})\mathcal{K}(\mathbf{X}, \mathbf{Y}) = \pi(\mathbf{Y})\mathcal{K}(\mathbf{Y}, \mathbf{X}), \ \forall \, \mathbf{X}, \mathbf{Y}. \tag{62}$$

The detailed balance equation is a sufficient condition for $\mathcal{K}$ satisfying the invariant condition,

$$\sum_{\mathbf{X}} \pi(\mathbf{X})\mathcal{K}(\mathbf{X}, \mathbf{Y}) = \pi(\mathbf{Y}), \ \forall \mathbf{Y}. \tag{63}$$

In practice, one may design a set of Markov chain kernels, each corresponding to a specific MCMC dynamics,

$$\Delta = \{\mathcal{K}_a, a \in \mathcal{A}\}, \tag{64}$$

The overall Markov chain kernel is a mixture of these dynamics with probability q_a,

$$\mathcal{K}(\mathbf{X}, \mathbf{Y}) = \sum_{a \in \mathcal{A}} q_a \mathcal{K}_a(\mathbf{X}, \mathbf{Y}), \ \ \forall \, \mathbf{X}, \mathbf{Y}. \tag{65}$$

There are two basic design criteria for Δ, which are easily observed in the finite state space.

1. The Kernels in Δ are ergodic so that for any two points $\mathbf{X}$ and $\mathbf{Y}$, there is a path of finite length $(\mathbf{X}, \mathbf{X}_1, ..., \mathbf{X}_N, \mathbf{Y})$ between $\mathbf{X}$ and $\mathbf{Y}$ consisting of the $N+1$ kernels $k(0), ..., K(N) \in \Delta$, with

$$\mathcal{K}_{k(0)}(\mathbf{X}, \mathbf{X}_1) \cdot \mathcal{K}_{k(1)}(\mathbf{X}_1, \mathbf{X}_2) \cdots \mathcal{K}_{k(N)}(\mathbf{X}_N, \mathbf{Y}) > 0.$$

2. Each sub-kernel observes the detailed balance equations, and thus the overall kernel satisfies them.

The multigrid and multi-level design in the next two subsections are ways for designing the sub-kernels that observe the detailed balance equations.

6.2 Multigrid clustering sample

Let Λ be an "attention window" on graph $\mathbf{G}$, and $\mathbf{X} = (V_1, V_2, ..., V_{\mathrm{L}})$ the current labeling state. Λ divides the vertices into two parts,

$$V = V_\Lambda \ \cup \ V_{\bar{\Lambda}}, \ \text{ and } \ \mathbf{X} = (\mathbf{X}_\Lambda, \mathbf{X}_{\bar{\lambda}}). \tag{66}$$

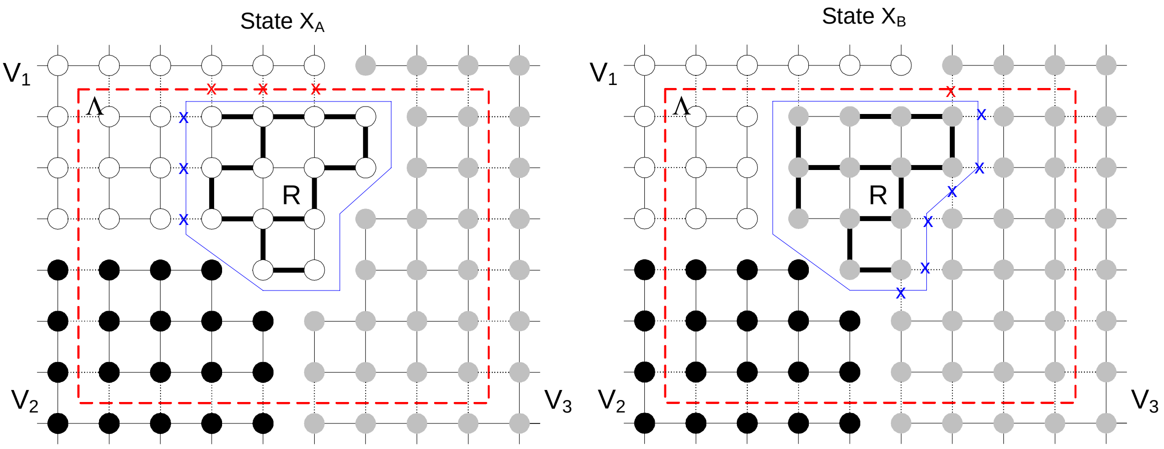


Figure 12: Multigrid flipping: computation is restricted to different "attention" windows Λ of various sizes, with the rest of the labels fixed.

For example, Fig.12 displays a rectangular window Λ (in red dashed) in a lattice $\mathbf{G}$. The window Λ cuts some edges within each subset $V_k, k = 1, 2, ..., \mathrm{L}$, and we denote them by,

$$\mathcal{C}(V_k, \Lambda) = \{< s, t >: s \in V_k \cap V_\Lambda,\ t \in V_k \cap V_{\bar{\Lambda}}\}.$$

In Fig.12 the window Λ intersects with three subsets V_1 (white), V_2 (black), and V_3 (grey), and all edges crossing the (red) rectangle window are cut.

multi-grid cluster sampling

1. Select an attention window $\Lambda \subset \mathbf{G}$.
2. Cluster the vertices within Λ and select connected component R.
3. Flip the label of R.
4. Accept the flipping with probability uses $\mathbf{X}_{\bar{\Lambda}}$ as the boundary condition.

Following the same procedure as in Section (3), we can derive the proposal probability ratio for selecting R in the two states $\mathbf{X}_A$ and $\mathbf{X}_B$ within Λ.

Proposition 9 *The probability ratio for proposing R as a candidate cluster within window Λ at two states $\mathbf{X}$ and $\mathbf{X}'$ is*

$$\frac{q(R|\mathbf{X}, \Lambda)}{q(R|\mathbf{X}', \Lambda)} = \frac{\prod_{<i,j>\in\mathcal{C}(R,V_\ell)-\mathcal{C}(V_\ell,\Lambda)}(1-q_{ij})}{\prod_{<i,j>\in\mathcal{C}(R,V_{\ell'})-\mathcal{C}(V_{\ell'},\Lambda)}(1-q_{ij})}.$$

In Fig. 12), we have $\mathbf{X} = \mathbf{X}_A$ and $\mathbf{X}' = \mathbf{X}_B$ ($\ell = 1, \ell' = 3$).

The difference between this ratio and the ratio in proposition 3 is that some edges in $\mathcal{C}(V_\ell, \Lambda) \cup \mathcal{C}(V_{\ell'}, \Lambda)$ no longer participate in the computation.

Proposition 10 *The Markov chain simulated by the multi-grid scheme has invariant probability* $\pi(\mathbf{X}_\Lambda|\mathbf{X}_{\bar{\Lambda}})$ *and its kernel* $\mathcal{K}$ *observes the detailed balance equation,*

$$\pi(\mathbf{X}_\Lambda|\mathbf{X}_{\bar{\Lambda}})\mathcal{K}(\mathbf{X}_\Lambda, \mathbf{Y}_\Lambda) = \pi(\mathbf{Y}_\Lambda|\mathbf{X}_{\bar{\Lambda}})\mathcal{K}(\mathbf{Y}_\Lambda, \mathbf{X}_\Lambda). \tag{67}$$

Proposition 11 *Let* $\pi(\mathbf{X})$ *be a target probability defined on a graph* $\mathbf{G} =< V, E >$ *and* $\Lambda \subset V$ *a window, if a cluster sampling step has a kernel* $\mathcal{K}_a$ *that observes the detailed balance equation with respect to a conditional probability, then it observes the detailed balance equations,*

$$\pi(\mathbf{X}_\Lambda)\mathcal{K}(\mathbf{X}_\Lambda, \mathbf{Y}_\Lambda) = \pi(\mathbf{Y}_\Lambda)\mathcal{K}(\mathbf{Y}_\Lambda, \mathbf{X}_\Lambda), \tag{68}$$

The proofs for the two propositions are straightforward.

6.3 Multi-level cluster sampling

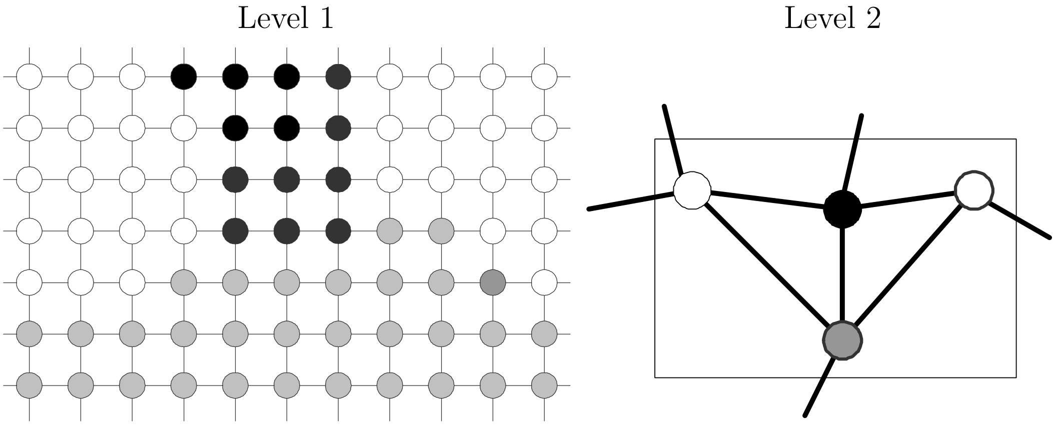


Figure 13: Multi-level cluster sampling. Computation is performed at different levels of granularity, where the connected components from the lower level collapse into vertices in the higher level.

Following the notations in Section (6.1), the problem is hierarchic labeling with $\mathbf{G} = (\mathbf{G}^{(0)}, \mathbf{G}^{(1)}, \mathbf{G}^{(2)})$ and $\mathbf{X} = (\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)})$. Each level of labeling $\mathbf{X}^{(s)}$ is equivalent to a partition of the lattice with connected components.

$$\mathrm{CP}(\mathbf{X}^{(s)}) = \{\mathrm{cp}_1^{(s)}, \mathrm{cp}_2^{(s)}, ..., \mathrm{cp}_{m^{(s)}}^{(s)}\},\ s = 0, 1, 2. \tag{69}$$

Note that vertices in each connected component have the same label and two disconnected components may share the same label.

Definition 2 *The hierarchic labels* $\mathbf{X} = (\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ *are said to be* "nested" *if*

$$\forall \mathrm{cp}^{(s)} \in \mathrm{CP}(\mathbf{X}^{(s)}),\ \exists \mathrm{cp}^{(s+1)} \in \mathrm{CP}(\mathbf{X}^{(s+1)}) \text{ so that } \mathrm{cp}^{(s)} \subset \mathrm{cp}^{(s+1)}, \quad s = 0, 1.$$

A nested $\mathbf{X}$ has a tree structure for the levels of labels. A vertex in level $s+1$ has a number of children vertices in level s.

multi-level cluster sampling

1. Select a level s, usually in an increasing order.
2. Cluster the vertices in $\mathbf{G}^{(s)}$ and select connected component R.
3. Flip the labeling of R.
4. Accept the flipping with probability uses the other levels (denoted by $\mathbf{X}^{(-s)}$) as the boundary condition.

Proposition 12 *Let* $\pi(\mathbf{X})$ *be a target probability with nested labeling* $\mathbf{X} = (\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)})$*, the cluster sampling on the three levels of graphs have kernels* $\mathcal{K}^{(0)}$*,* $\mathcal{K}^{(1)}$ *and* $\mathcal{K}^{(2)}$ *respectively. If they observe the detailed balance equations with respect to the conditional probabilities,*

$$\pi(\mathbf{X}^{(s)}|\mathbf{X}^{(-s)})\mathcal{K}^{(s)}(\mathbf{X}^{(s)}, \mathbf{Y}^{(s)}) = \pi(\mathbf{Y}^{(s)}|\mathbf{X}^{(-s)})\mathcal{K}^{(s)}(\mathbf{Y}^{(s)}, \mathbf{X}^{(s)}),\ s = 0, 1, 2. \tag{70}$$

where $\mathbf{X}^{(-s)}$ *means* $\mathbf{X}$ *except* $\mathbf{X}^{(s)}$*, then it observes the detailed balance equation (62).*

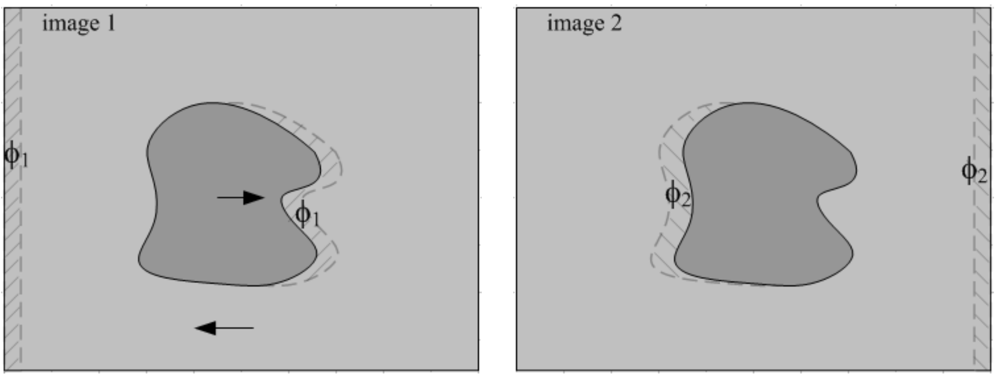


image frame $\mathbf{I}_1$ image frame $\mathbf{I}_2$

Figure 14: Two consecutive image frames with two moving objects in the foreground and background respectively. The pixels in area ϕ_1 are not seen in $\mathbf{I}_2$ and reversely the pixels in ϕ_1 are not seen in image $\mathbf{I}_1$, and they are called "half-occluded" images. Other pixels can be mapped between the two frames. The displacement stands for the planar motion.

7 Experiment 2: hierarchic motion segmentation

Now we report the experiments on motion analysis using multi-grid and multi-level cluster sampling.

Let $\mathbf{I}_1, \mathbf{I}_2$ be two consecutive image frames in a video as Fig. 14 illustrates, due to motion occlusion, some points are visible in only one image, say the shadow areas ϕ_1 in $\mathbf{I}_1$ and ϕ_2 in $\mathbf{I}_2$ which are called "half-occluded" points, and all other points can be mapped between the two image frames ρ_1 and ρ_2. The mapping function is called the "optical flow" field,

$$(u, v) : \rho_2 \backslash \phi_2 \mapsto \rho_1 \backslash \phi_1. \qquad (71)$$

For any point (x, y) in the first frame, $(u(x, y), v(x, y))$ is the displacement for the planar motion velocity. Usually one can assume that the intensity of a point will be constant (with stable illumination and Lambertian surfaces) between two frames, and the residue is modeled by Gaussian noise $\mathbf{n} \sim \text{Gaussian}(0, \sigma_o^2)$. Let's take the second image as the reference frame,

$$\mathbf{I}_2(x, y) = \mathbf{I}_1(x - u(x, y), y - v(x, y)) + \mathbf{n}(x, y), \;\; \forall\, (x, y) \in \rho_2 \backslash \phi_2. \qquad (72)$$

We discritize the image planes ρ_1 and ρ_2 into lattices Λ_1 and Λ_2 respectively. In the motion analysis problem, we consider discrete pixels in the second image frame $\mathbf{G}^{(0)} = \Lambda_2$, and each pixel has three labels $x = (x^{(0)}, x^{(1)}, x^{(2)})$.

1. Its velocity $x^{(0)} = (u, v)$ which is discretized into $13 \times 13 = 169$ different planar velocities. We assume the maximum displacement in the lattice between two consecutive frames to be $-3 \leq u, v \leq 3$ with 1/2 pixel precision. That leads to 169 possible planar velocities. Then for pixels which do not have corresponding pixels in the first frame, i.e. pixels in ϕ_2, their velocities cannot be decided and denote it by nil. It can be estimated based on context information on their intensity through image segmentation. Thus we have $x^{(0)} \in \{\text{nil}, 1, 2, ..., 169\}$ as its velocity label.

2. Its intensity label $x^{(1)} \in \{1, 2, ..., \mathrm{L}^{(1)}\}$ for image segmentation. That is, the image lattice is partitioned into a number of regions with coherent intensities in terms of fitting to the three families of image models in Section (5).

3. Its object label $x^{(2)} \in \{1, 2, ..., \mathrm{L}^{(2)}\}$. That is, the image lattice is partitioned into a number of $\mathrm{L}^{(2)}$ objects which have coherent intensity and motion.

To fix notation, we divide the image frames into two parts,

$$\mathbf{I}_1 = (\mathbf{I}_{1,\phi_1}, \mathbf{I}_{1,\bar{\phi}_1}), \quad \mathbf{I}_2 = (\mathbf{I}_{2,\phi_2}, \mathbf{I}_{2,\bar{\phi}_2})$$

The target probability is the Bayesian posterior,

$$\pi(\mathbf{X}) = \pi(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)} | \mathbf{I}_1, \mathbf{I}_2) \propto \mathcal{L}(\mathbf{I}_{1,\bar{\phi}_1} | \mathbf{I}_{2,\bar{\phi}_2}, \mathbf{X}^{(0)}) \mathcal{L}(\mathbf{I}_2 | \mathbf{X}^{(1)}) \pi_o(\mathbf{X}). \tag{73}$$

The first likelihood is specified by the optical flow model,

$$\mathcal{L}(\mathbf{I}_{1,\bar{\phi}_1} | \mathbf{I}_{2,\bar{\phi}_2}, \mathbf{X}^{(0)}) = \prod_{(x,y) \in \Lambda_2 \setminus \phi_2} \frac{1}{\sqrt{2\pi}\sigma_o} \exp\{-\frac{1}{2\sigma_o}(\mathbf{I}_2(x,y) - \mathbf{I}_1(x - u(x,y), y - v(x,y)))^2\}. \tag{74}$$

The second likelihood is the same as the image segmentation likelihood in Section (5). The prior probability assumes piecewise coherent motion. That is, each moving object $o = 1, 2, ..., \mathrm{L}^{(2)}$ has a constant planar velocity $c_o \in \{1, 2, ..., 169\}$ plus a Markov model for the adjacent velocities. Also each object (and region) has compact boundary.

$$
\begin{aligned}
\pi_o(\mathbf{X}) \quad & \propto \prod_{o=1}^{\mathrm{L}^{(2)}} \exp\{-\alpha \sum_{v, x^{(2)}(v)=o} |x^{(0)}(v) - c_o|^2 - \beta \sum_{v' \in \partial v} |x^{(0)}(v') - x^{(0)}(v)|\} \\
& \prod_{l=1}^{\mathrm{L}^{(1)}} \exp\{-\gamma |\partial V_l^{(1)}|\} \prod_{i=1}^{\mathrm{L}^{(0)}} \exp\{-\delta |\partial V_i^{(0)}|\} \exp\{-\lambda_0 \mathrm{L}^{(0)} - \lambda_1 \mathrm{L}^{(1)} - \lambda_2 \mathrm{L}^{(2)}\} \qquad (75)
\end{aligned}
$$

Now we define the edge probability at the three levels of graph for the auxiliary variables.

At level $\mathbf{X}^{(0)}$, let (x, y) and (x', y') be two adjacent pixels, and (u, v) the common motion velocity of both pixels, The edge probability is defined as

$$
\begin{aligned}
q^{(0)}(v, v') \quad &= \quad \min_{(u,v)} \exp\{-[|\mathbf{I}_2(x, y) - \mathbf{I}_1(x - u, y - v)| + |\mathbf{I}_2(x', y') - \mathbf{I}_1(x' - u, y' - v)|]/7 \\
&= \quad -|\mathbf{I}_2(x, y) - \mathbf{I}_2(x', y')|/10\}.
\end{aligned}
$$

At the region level $\mathbf{X}^{(1)}$, the edge weights between two adjacent nodes v, v' (each being a set of pixels) are based on the KL divergence between their intensity histograms h_u, h_v, as in Section 5.

At the object level $\mathbf{X}^{(2)}$, the edge weights between two adjacent nodes v, v' (each being a set of pixels) are based on the KL divergence between their motion histograms $h_m(v), h_m(v')$. We maintain the histogram of the motion velocities in each object.

$$
q^{(2)}(v, v') = \exp\{-\frac{1}{2}(KL(h_m(v)||h_m(v')) + KL(h_m(v')||h_m(v))\}. \qquad (76)
$$

We run the multi-grid and multi-level SW-cut on a number of synthetic and real world motion images. We show four results in Fig.15. The first image shows two moving rectangles where only the 8 corners provide reliable local velocity (aperture problem) and the image segmentation is instrumental in deriving the right result. For the other three sequences, the algorithm obtains satisfactory results despite large motion and complex background. The cheetah image in Fig.11 is a fifth example.

We choose the segmentation example – the cheetah image in Fig. 7 for comparison of the different cluster sampling methods. In section (5), the pixels are grouped deterministically into atomic regions in a pre-processing stage. Now we do the cluster sampling in two levels and the atomic regions are generated by one level of the cluster sampling process.

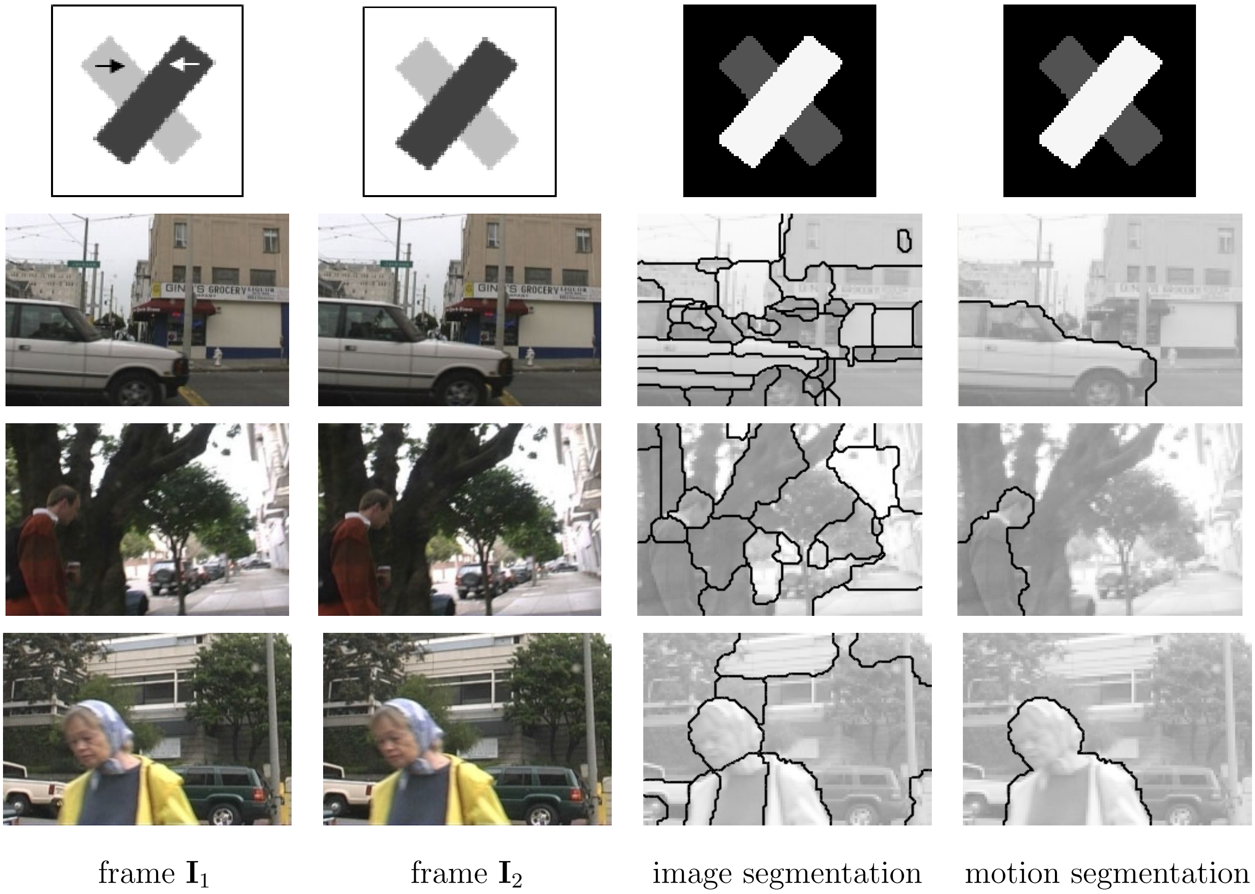


Figure 15: Hierarchical motion analysis. From left to right: first frame $\mathbf{I}_1$, second frame $\mathbf{I}_2$, image segmentation, motion segmentation. The image segmentation is the result at level $s = 1$ and the motion segmentation is the result at level $s = 2$. For the color images (the 3rd and 4th rows) we treated the three R,G, B color bands each as a grey image.

We plot in Fig.**??** the $-\ln \pi(\mathbf{X})$ vs the CPU time for various methods. This figure should be compared with Fig. 10. The multi-level cluster sampling was run in two initializations.

Firstly, the two level cluster sampling is much slower than the the one level clustering. The latter assumed deterministic atomic regions. But the two level cluster sampling can reach a deeper minimum as it has more flexibility in forming the atomic regions.

Secondly, the multi-grid method is the fastest among the methods that work directly on pixels.

Thirdly, the Gibbs sampler plotted in Fig. 10 run on the deterministic atomic regions not the pixels. If it is running on the pixels, we cannot get it converge to the minimum in

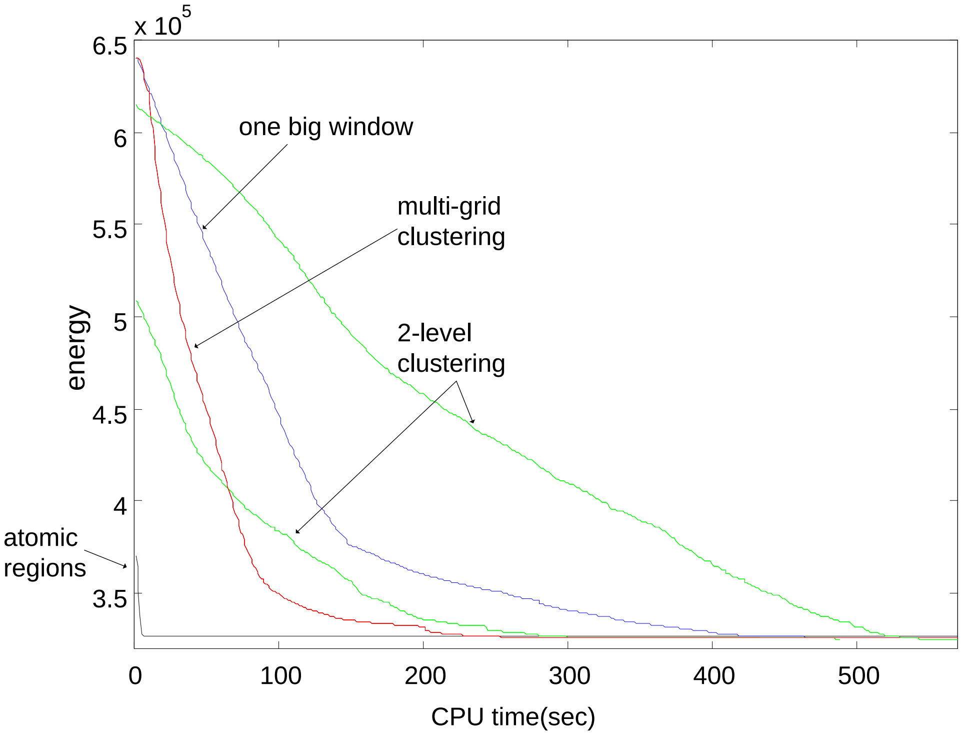


Figure 16: Convergence comparison of multigrid and multi-level cluster sampling for the cheetah image in Fig. 7. (see text for explanation)

any short time.

8 Discussion

In this paper, we only report the empirical speed of the cluster sampling methods. In the literature, there are no analytic results for even the original Swendsen-Wang method in the presence of external fields, for it is difficult in quantifying the external fields. In our case, it is impractical to quantify the natural images with a reasonable model. In our experiment, the cluster Gibbs sampler with acceptance probability 1 does not necessarily mix faster than the cluster sampling with rejection. The former is more computationally costly in each step. These problems remain open for further investigation.

References

[1] Barbu, A. and Zhu, S.C. (2003).“Graph partition by Swendsen-Wang cuts”, *Proc. Int'l Conf. on Computer Vision*, Nice, France.

[2] Barbu, A. and Zhu, S.C. (2004). “Multigrid and multi-level Swendsen-Wang cuts for hierarchic graph partition”, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Washington DC, 2004.

[3] Cooper, C. and Frieze, A. (1999). “Mixing properties of the Swendsen-Wang process in classes of graphs”, *Random Structures and Algorithms* **15**, no. 3-4, 242-261.

[4] Edwards, R.G. and Sokal, A.D. (1988). “Generalization of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm”, *Phys. Rev. Lett.* **38**, 2009-2012.

[5] Geman, S. and Geman, D. (1984),“Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images”, *IEEE Trans. on PAMI* **6**, 721-741.

[6] Gilks, W.R. and Roberts, G. O. (1996). ”Strategies for improving MCMC”, in (Gilks, W.R. eds) *Markov Chain Monte Carlo in Practice*, Chapman & Hall/CRC .

[7] Gore, V. and Jerrum, M (1997). “The Swendsen-Wang process does not always mix rapidly”, *Proc. 29th ACM Symp. on Theory of Computing* 674-681.

[8] Green, P. J. (1995). “Reversible jump MCMC comput. and Bayes. model determination”,*Biometrika*,**82**, 711-732.

[9] Hastings, W.K. (1970). “Monte Carlo sampling methods using Markov chains and their applications”, *Biometrika* **57**, 97-109.

[10] Higdon, D.M. (1998). “Auxiliary variable methods for Markov chain Monte Carlo with applications”, *J. Am. Statist. Assoc.* **93**, 585-595.

[11] Huber, M. (2002). “A bounding chain for Swendsen-Wang.” *Random Structures and Algorithms* **22**, no 1, 43-59.

[12] Ising, E (1925). “Beitrag zur theorie des ferromagnetismus”, *Zeitschrift für Physik* **31**, 253-258.

[13] Liu, J.S. and Wu, Y.N. (1999). “Parameter expansion scheme for data augmentation”, *J. Am. Statist. Assoc.* **94**.

[14] Liu, J.S. (2001). “Monte Carlo strategies in scientific computing”, Springer, NY.

[15] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. (1953). “Equations of the state calculations by fast computing machines”, *J. Chem. Physics* **22**, 1087-1091.

[16] Potts, R.B. (1953) “Some generalized order-disorder transformations”, *Proceedings of the Cambridge Philosophic Society* **48**, 106-109.

[17] Swendsen, R.H. and Wang, J.S. (1987), “Nonuniversal critical dynamics in Monte Carlo simulations”, *Physical Review Letters* **58** no. 2, 86-88.

[18] Tanner, M. A. and Wong, W.H. (1987), ”The calculation of posterior distributions by data augmentation (with discussion)”, *J. Amer. Stat. Assoc.*, 82(398):528-540.

[19] Tu, Z.W. and Zhu, S.C. (2002). “Image segmentation by data-driven Markov chain Monte Carlo”, *IEEE Trans. on PAMI* **24**, no. 5.

[20] Wolff, U. (1989). “Collective Monte Carlo updating for spin systems”, *Physical Review Letters* **62**, no. 4, 361-364.

Appendix A Proof of Proposition 3

Consider a reversible jump between two states $\mathbf{X}$ and $\mathbf{X}'$ which differ only in the labeling of R,

$$\mathbf{X}_R = \ell \neq \ell' = \mathbf{X}'_R, \quad \mathbf{X}_{\bar{R}} = \mathbf{X}_{\bar{R}}. \tag{77}$$

Our objective is to derive the proposal probability ratio $\frac{q(R|\mathbf{X})}{q(R|\mathbf{X}')}$ for selecting R in $\mathbf{X}$ and $\mathbf{X}'$. This ration depends on the probabilities in the clustering and flipping steps.

Let $\mathbf{U}|\mathbf{X}$ and $\mathbf{U}'|\mathbf{X}'$ be the auxiliary variables following the Bernoulli probabilities in the flipping step, and they leads to two sets of connected components $\mathrm{CP}(\mathbf{U}|\mathbf{X})$ and $\mathrm{CP}(\mathbf{U}'|\mathbf{X}')$ respectively. We divide $\mathbf{U}$ into two sets for the on and off edges respectively,

$$\mathbf{U} = \mathbf{U}_{\text{on}} \cap \mathbf{U}_{\text{off}}. \tag{78}$$

$$\mathbf{U}_{\text{on}} = \{\mu_{ij} \,:\, \mu_{ij} = 1\}, \;\; \mathbf{U}_{\text{off}} = \{\mu_{ij} \,:\, \mu_{ij} = 0\}.$$

We are only interested in the $\mathbf{U}$'s (and thus CP's) which yield the connected component R. We collect all such $\mathbf{U}$ given $\mathbf{X}$ in a set,

$$\Psi(R|\mathbf{X}) = \{\mathbf{U} \,:\, R \in \mathrm{CP}(\mathbf{U}|\mathbf{X})\}. \tag{79}$$

In order for R being a connected component in $\mathbf{X}$, all edges between R and $V_\ell \backslash R$ must be cut (turned off), otherwise R is connected to other vertices in V_ℓ and can not be a connected component. So, we denote the remaining "off" edges by ${}^{-}\mathbf{U}_{\text{off}}$,

$$\mathbf{U}_{\text{off}} = \mathcal{C}(R, V_\ell) \,\cup\, {}^{-}\mathbf{U}_{\text{off}}, \quad \forall\, \mathbf{U} \in \Psi(R|\mathbf{X}). \tag{80}$$

Similarly, we collect all $\mathbf{U}'$ in state $\mathbf{X}'$ which produce the connected component R,

$$\Psi(R|\mathbf{X}') = \{\mathbf{U}' \,:\, R \in \mathrm{CP}(\mathbf{U}'|\mathbf{X}')\}. \tag{81}$$

In order for R to be a connected component in $\mathbf{U}'|\mathbf{X}'$, the clustering step must cut all the edges between R and $V_{\ell'}$. Thus we have

$$\mathbf{U}' = \mathbf{U}'_{\text{on}} \cap \mathbf{U}'_{\text{off}} \tag{82}$$

with

$$\mathbf{U}'_{\text{off}} = \mathcal{C}(R, V_{\ell'}) \,\cup\, {}^{-}\mathbf{U}'_{\text{off}}, \quad \forall\, \mathbf{U}' \in \Psi(R|\mathbf{X}'). \tag{83}$$

A key observation is that there is a one-to-one mapping between $\Psi(R|\mathbf{X})$ and $\Psi(R|\mathbf{X}')$.

Proposition 13 *For any $\mathbf{U} \in \Psi(R|\mathbf{X})$, there exists one and only one $\mathbf{U}' \in \Psi(R|\mathbf{X}')$ such that*

$$\mathrm{CP}(\mathbf{U}|\mathbf{X}) = \mathrm{CP}(\mathbf{U}'|\mathbf{X}') \tag{84}$$

and

$$\mathbf{U}_{\text{on}} = \mathbf{U}'_{\text{on}},{}^{-}\,\mathbf{U}_{\text{off}} ={}^{-}\,\mathbf{U}'_{\text{off}}. \tag{85}$$

That is, $\mathbf{U}$ and $\mathbf{U}'$ differ only in the cuts $\mathcal{C}(R, V_\ell)$ and $\mathcal{C}(R, V_{\ell'})$.

Suppose that we choose $R \in \mathrm{CP}$ with probability $q(R|\mathrm{CP})$, the probability for choosing R at $\mathbf{X}$ is the sum over all possible $\mathbf{U} \in \Psi(R|\mathbf{X})$ with the probability of choosing $\mathbf{U} \in \Psi(R|\mathbf{X})$ times the probability of choosing R from $\mathrm{CP}(\mathbf{U}|\mathbf{X})$,

$$q(R|\mathbf{X}) = \sum_{\mathbf{U} \in \Psi(R|\mathbf{X})} [q(R|\mathrm{CP}(\mathbf{U}|\mathbf{X})) \prod_{<i,j> \in \mathbf{U}_{\mathrm{on}}} q_{ij} \prod_{<i,j> \in^{-} \mathbf{U}_{\mathrm{off}}} (1-q_{ij})] \prod_{<i,j> \in \mathcal{C}(R,V_\ell)} (1-q_{ij}). \quad (86)$$

Similarly, the probability for choosing $R \subseteq V_{\ell'}$ at $\mathbf{X}'$ is

$$q(R|\mathbf{X}') = \sum_{\mathbf{U}' \in \Psi(R|\mathbf{X}')} [q(R|\mathrm{CP}(\mathbf{U}'|\mathbf{X}')) \prod_{<i,j> \in \mathbf{U}'_{\mathrm{on}}} q_{ij} \prod_{<i,j> \in^{-} \mathbf{U}'_{\mathrm{off}}} (1-q_{ij})] \prod_{<i,j> \in \mathcal{C}(R,V_{\ell'})} (1-q_{ij}). \quad (87)$$

Dividing eqn. (86) by eqn. (87), we obtain the ratio in eqn. (39) due to cancelation following the observations in Proposition 13.

$$\frac{q(R|\mathbf{X})}{q(R|\mathbf{X}')} = \frac{\prod_{<i,j> \in \mathcal{C}(R,V_\ell)} (1-q_{ij})}{\prod_{<i,j> \in \mathcal{C}(R,V_{\ell'})} (1-q_{ij})}. \quad (88)$$

In a special case when $R = V_\ell$, then $\mathcal{C}(R, V_\ell) = \emptyset$ and $\prod_{<i,j> \in \mathcal{C}(R,V_\ell)} (1-q_{ij}) = 1$.

End of Proof.

Note that the proof holds for arbitrary design of q_{ij}, arbitrary design of $q(R|\mathrm{CP}(\mathbf{U}|\mathbf{X}))$ on arbitrary graphs. When the graph is very densely connected, then the cuts $\mathcal{C}(R, V_\ell)$ and $\mathcal{C}(R, V_{\ell'})$ will become large. For the graphs with $O(1)$ connectivity as in the image applications, the sizes of the cuts $\mathcal{C}(R, V_\ell)$ are in the order of the perimeter if the component R, i.e. $O(|\partial R|)$.

Appendix B Proof of Theorem 4

[Proof] For the canonical case, there is a unique path moving between $\mathbf{X}$ and $\mathbf{X}'$ in one step – choosing R and changing its label. Therefore we rewrite eqn.(38),

$$\frac{q(\mathbf{X} \to \mathbf{X}')}{q(\mathbf{X}' \to \mathbf{X})} = \frac{q(R|\mathbf{X})}{q(R|\mathbf{X}')} \cdot \frac{q(\mathbf{X}_R = \ell'|R, \mathbf{X})}{q(\mathbf{X}_R = \ell|R, \mathbf{X}')}. \quad (89)$$

Plug it in the Metropolis-Hastings eqn.(36), we obtain the result.

For the split and merge cases (see Section 3.2), there are two paths moving between $\mathbf{X}$ and $\mathbf{X}'$ in one step. The proposal probability is the sum of proposal probabilities in the two pathes.

Without loss of generality, let $\mathbf{X} = (V_1, V_2, V_3, ..., V_n)$ and $\mathbf{X}' = (V_{1+2}, V_3, V_4, ..., V_n)$ with $V_{1+2} = V1 \cup V_2$.

- Path 1: Choose $R = V_1$ in $\mathbf{X}$ and merge it to V_2 (i.e. choosing $\mathbf{X}_R = 2$) to reach $\mathbf{X}'$, and reversely, Choose $R = V_1 \subset V_{1+2}$ in $\mathbf{X}'$ and split it to a new color V_1 (i.e. choosing $\mathbf{X}_R = 1$) and the rest $V_{1+2} \backslash V_1$ is named V_2.
- Path 2: Choose $R = V_2$ in $\mathbf{X}$ and merge it to V_1 (i.e. choosing $\mathbf{X}_R = 1$) to reach $\mathbf{X}'$, and reversely, Choose $R = V_2 \subset V_{1+2}$ in $\mathbf{X}'$ and split it to a new color V_2 and the rest $V_{1+2} \backslash V_2$ is named V_1.

$$\frac{q(\mathbf{X} \to \mathbf{X}')}{q(\mathbf{X}' \to \mathbf{X})} = \frac{q(R = V_1|\mathbf{X})q(\mathbf{X}_R = 2|R = V_1, \mathbf{X}) + q(R = V_2|\mathbf{X})q(\mathbf{X}_R = 1|R = V_2, \mathbf{X})}{q(R = V_1|\mathbf{X}')q(\mathbf{X}_R = 1|R = V_1, \mathbf{X}') + q(R = V_2|\mathbf{X}')q(\mathbf{X}_R = 2|R = V_2, \mathbf{X}')}. \tag{90}$$

Then we have two observations in the following.

Firstly, from Proposition 3, we know,

$$\frac{q(R = V_1|\mathbf{X})}{q(R = V_1|\mathbf{X}')} = \frac{1}{\prod_{<i,j> \in \mathcal{C}(V_1, V_2)}(1 - q_{ij})} = \frac{q(R = V_2|\mathbf{X})}{q(R = V_2|\mathbf{X}')} \tag{91}$$

Secondly, once R is selected from $\mathbf{X}$ (or $\mathbf{X}'$), its new label follows a label proposal probability which depends on the partition of all other vertices $V \backslash R$ which are the same for both $\mathbf{X}$ and $\mathbf{X}'$. Note that all permutations of the labelings are considered equivalent. Therefore we have

$$\frac{q(\mathbf{X}_R = 2|R = V_1, \mathbf{X})}{q(\mathbf{X}_R = 1|R = V_1, \mathbf{X}')} = \frac{q(\mathbf{X}_R = 1|R = V_2, \mathbf{X})}{q(\mathbf{X}_R = 2|R = V_2, \mathbf{X}')}. \tag{92}$$

Therefore, we can write the ratio in both paths as $\frac{q(\mathbf{X}_R = \ell'|R, \mathbf{X})}{q(\mathbf{X}_R = \ell|R, \mathbf{X}')}$. Plug eqns. (91) and (92) in eq. (90), we have the result.

The split case is the reverse of the merge case and thus both cases are proven in the above discussion. *End of proof*

Appendix C Proof of Proposition 7

[Proof]Let the label of R in state $\mathbf{X}$ be $\mathbf{X}_R = \ell$ and after relabeling $\mathbf{X}'_R = \ell'$. By the Metropolis acceptance eqn. (36) and by (38) and Prop. 3, we obtain

$$\alpha(\mathbf{X} \to \mathbf{X}') = \min\{1, \frac{\gamma_{\ell'}}{\gamma_\ell} \cdot \frac{q(\mathbf{X}_R = \ell|R, \mathbf{X}')}{q(\mathbf{X}_R = \ell'|R, \mathbf{X})} \cdot \frac{\pi(\mathbf{X}')}{\pi(\mathbf{X})}\}. \tag{93}$$

We observe that the number and values of γ_k do not depend on the particular value of $\mathbf{X}_R$, so in both states $\mathbf{X}, \mathbf{X}'$, all γ_k are the same. Since $\mathbf{X}_{\partial R} = \mathbf{X}'_{\partial R}$, we have

$$\sum_{k=1}^{N(\mathbf{X})} \gamma_k \cdot \pi(\mathbf{X}_R = k|\mathbf{X}_{\partial R}) = \sum_{k=1}^{N(\mathbf{X}')} \gamma_k \cdot \pi(\mathbf{X}'_R = k|\mathbf{X}'_{\partial R}) \tag{94}$$

so

$$\frac{q(\mathbf{X}_R = \ell|R, \mathbf{X}')}{q(\mathbf{X}_R = \ell'|R, \mathbf{X})} = \frac{\gamma_\ell \cdot \pi(\mathbf{X})}{\gamma_{\ell'} \cdot \pi(\mathbf{X}')} \tag{95}$$

So we get

$$\alpha(\mathbf{X} \to \mathbf{X}') = \min\{1, \frac{\gamma_{\ell'}}{\gamma_\ell} \cdot \frac{\gamma_\ell \cdot \pi(\mathbf{X})}{\gamma_{\ell'} \cdot \pi(\mathbf{X}')} \cdot \frac{\pi(\mathbf{X}')}{\pi(\mathbf{X})}\} = 1, \tag{96}$$

which means the move is always accepted. *End of proof*

Appendix D Proof of Proposition 8

[Proof] We will proceed in a similar fashion with the proof of Prop. 3, from Apendix A. We maintain the notations for $\mathbf{U}_{\text{on}}, \mathbf{U}_{\text{off}}, \text{CP}(\mathbf{U}|\mathbf{X})$ from Appendix A.

In state $\mathbf{X}$, let $\mathbf{U}$ be one of the many sets of auxiliary variables that can be used to obtain the connected components $D(\mathbf{X}, \mathbf{X}')$. Then any cp $\in D(\mathbf{X}, \mathbf{X}')$ is connected through edges of $\mathbf{U}_{\text{on}}$.The probability to obtain state $\mathbf{X}'$ through flipping the components from $\text{CP}(\mathbf{U}|\mathbf{X})$ independently is

$$q(\mathbf{X}'|\mathbf{U}, \mathbf{X}) = \prod_{\text{cp} \in \text{CP}(\mathbf{U}|\mathbf{X})} q(\mathbf{X}'|\text{cp}) \tag{97}$$

The probability to go from state $\mathbf{X}$ to $\mathbf{X}'$ is

$$q(\mathbf{X}'|\mathbf{X}) = \sum_{\mathbf{U}} \prod_{\text{cp} \in \text{CP}(\mathbf{U}|\mathbf{X})} q(\mathbf{X}'|\text{cp}) \prod_{<i,j> \in \mathbf{U}_{\text{on}}} q_{ij} \prod_{<i,j> \in \mathbf{U}_{\text{off}}} (1 - q_{ij}) \tag{98}$$

Let

$$^{-}\mathbf{U}_{\text{off}} = \mathbf{U}_{\text{off}} \backslash \mathcal{C}(\mathbf{X} \to \mathbf{X}') \tag{99}$$

Then

$$q(\mathbf{X}'|\mathbf{X}) = \prod_{<i,j>\in\mathcal{C}(\mathbf{X}\to\mathbf{X}')}(1-q_{ij})\prod_{\mathrm{cp}\in D(\mathbf{X},\mathbf{X}')}q(\mathbf{X}'|\mathrm{cp})\sum_{\mathbf{U}}\prod_{\mathrm{cp}\in\mathrm{CP}(\mathbf{U}|\mathbf{X})\setminus D(\mathbf{X},\mathbf{X}')}q(\mathbf{X}'|\mathrm{cp})\prod_{<i,j>\in\mathbf{U}_{\mathrm{on}}}q_{ij}\prod_{<i,j>\in^{-}\mathbf{U}_{\mathrm{off}}}(1-q_{ij}) \tag{100}$$

Similarly, the probability of going from state $\mathbf{X}'$ to $\mathbf{X}$ is

$$q(\mathbf{X}|\mathbf{X}') = \prod_{<i,j>\in\mathcal{C}(\mathbf{X}'\to\mathbf{X})}(1-q_{ij})\prod_{\mathrm{cp}\in D(\mathbf{X},\mathbf{X}')}q(\mathbf{X}|\mathrm{cp})\sum_{\mathbf{U}'}\prod_{\mathrm{cp}\in\mathrm{CP}(\mathbf{U}'|\mathbf{X}')\setminus D(\mathbf{X},\mathbf{X}')}q(\mathbf{X}|\mathrm{cp})\prod_{<i,j>\in\mathbf{U}'_{\mathrm{on}}}q_{ij}\prod_{<i,j>\in^{-}\mathbf{U}'_{\mathrm{off}}}(1-q_{ij}) \tag{101}$$

Smilarly to Apendix A, there is a one-to-one correspondence between auxiliary variables $\mathbf{U}$ in state $\mathbf{X}$ and $\mathbf{U}'$ in state $\mathbf{X}'$ such that such that

$$\mathrm{CP}(\mathbf{U}|\mathbf{X}) = \mathrm{CP}(\mathbf{U}'|\mathbf{X}') \tag{102}$$

and

$$\mathbf{U}_{\mathrm{on}} = \mathbf{U}'_{\mathrm{on}}, {}^{-}\mathbf{U}_{\mathrm{off}} = {}^{-}\mathbf{U}'_{\mathrm{off}}. \tag{103}$$

Then the sums in eqs. 100 and 101 are equal, so we obtain, by cancellation

$$\frac{q(\mathbf{X}|\mathbf{X}')}{q(\mathbf{X}'|\mathbf{X})} = \frac{\prod\limits_{<i,j>\in\mathcal{C}(\mathbf{X}\to\mathbf{X}')}(1-q_{ij})\prod\limits_{\mathrm{cp}\in D(\mathbf{X},\mathbf{X}')}q(\mathbf{X}'|\mathrm{cp})}{\prod\limits_{<i,j>\in\mathcal{C}(\mathbf{X}'\to\mathbf{X})}(1-q_{ij})\prod\limits_{\mathrm{cp}\in D(\mathbf{X},\mathbf{X}')}q(\mathbf{X}|\mathrm{cp})} \tag{104}$$

which, by applying the Metropolis acceptance eq. 36, gives the desired result. *End of proof*