.

Generalizing Swendsen-Wang for Image Analysis

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

[1] Siemens Corporate Research

755 College Rd. E

Princeton, NJ 08540

[2] University of California, Los Angeles

Department of Statistics

8125 Math Science Bldg, Box 951554

Los Angeles, CA 90095

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

Abstract

Markov chain Monte Carlo (MCMC) methods have been used in many fields (physics, chemistry, biology, and computer science) for simulation, inference, and optimization. In many applications, Markov chains are simulated for sampling from a target probabilities $\pi(\mathbf{X})$ defined on graphs $\mathbf{G}$. The graph vertices represent elements of the system, the edges represent spatial relationships while $\mathbf{X}$ is a vector of variables on the vertices which often take discrete values called labels or colors. Designing efficient Markov chains is a challenging task when the variables are strongly coupled. Because of this, methods such as the single-site Gibbs sampler often experience suboptimal performance. A well-celebrated algorithm, the Swendsen-Wang (1987) (SW) method, can address the coupling problem. It clusters the vertices as connected components after turning off some edges probabilistically, and changes the color of one cluster as a whole. It is known to mix rapidly under certain conditions. Unfortunately, the SW method has limited applicability and slows down in the presence of "external fields" e.g. likelihoods in Bayesian inference. In this paper, we present a general cluster algorithm which extends the SW algorithm to general Bayesian inference on graphs. We focus on image analysis problems where the graph sizes are in the order of $10^3 - 10^6$ with small connectivity. The edge probabilities for clustering are computed using discriminative probabilities from data. We design versions of the algorithm to work on multi-grid and multi-level graphs, and present applications to two typical problems in image analysis, namely image segmentation and motion analysis. In our experiments, the algorithm is at least two orders of magnitude faster (in CPU time) than the single-site Gibbs sampler.

Keywords: Swendsen-Wang, data augmentation, auxiliary variables, 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 a 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 $\mathbf{X} = (x_1, x_2, ..., x_n)$ defined on the vertices $V = \{v_1, v_2, ..., v_n\}$ of a graph $\mathbf{G} =< V, E >$, where the edges E represent spatial relationships and similarity between the nodes. 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 general graphs and target probabilities, we will focus on two examples in image analysis, namely image segmentation and motion analysis. For such applications, the graph $\mathbf{G}$ consists of image elements such as pixels and is very large, having $10^3 - 10^6$ vertices. At the same time, $\mathbf{G}$ is sparse, having constant $O(1)$ nearest neighbor connectivity that does not grow with the number of vertices. The state x_i is the color (or label) for image segmentation or discretized motion velocity for motion analysis.

In the literature, a generally applicable algorithm for Markov chain design is the Gibbs sampler (Geman and Geman 1984) and its generalizations, such as multi-grid (Goodman and Sokal 1989), parameter expansion (Liu and Wu 1999), simulated tempering (Geyer and Thompson 1995), and parallel tempering (Geyer 1991). These methods are sometimes slow, especially when strong coupling between the variables occurs. The strong coupling is frequent in image analysis because of strong prior models used to regularize the results.

One method that addresses the coupling problem is the Swendsen-Wang (1987) method for simulating the Ising/Potts models (Ising 1925, Potts 1953) in statistical physics. At each iteration, the Swendsen-Wang (SW) method forms a random graph by turning on/off the edges between vertices with identical labels through sampling Bernoulli variables defined on edges. Then it changes the color of all vertices of one cluster (connected component) in a single step.

The SW method is found to mix rapidly under certain conditions. For example, Cooper and Frieze (1999) show that SW has polynomial mixing time on the Ising/Potts models for graphs with $O(1)$ connectivity, even near the critical temperature. Gore and Jerrum (1997) showed that SW has exponential mixing time when $\mathbf{G}$ is a complete graph. Huber (2002) designed bounding chains for the SW method to diagnose exact sampling for high and low temperatures of the Potts model. The SW convergence can also be analyzed using a maximal correlation technique (Liu 2001, chapter 7). Despite its success, the power of the SW method is limited for two reasons.

1. It is only directly applicable to the Ising/Potts models and cannot be applied easily to arbitrary probabilities on general graphs (Higdon 1998).

2. It uses a constant probability for the binary variables on edges, and does not make use of the data information in clustering the vertices. Because of this, it slows down in the presence of "external fields" i.e. data (Higdon 1998).

In this paper, we present a general cluster 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 on the edges, computed from the input data, to measure the compatibility of the two adjacent vertices. Therefore the clustering step is informed by the data (external field) and leads to significant speedup empirically.

3. 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 at least two orders of magnitude faster than the Gibbs sampler (see Figs. 6, 7).

In the literature, there are two famous interpretations of the SW-method, namely the Random Cluster Model (RCM) interpretation of Edwards and Sokal (1988) and slice sampling intepretation of Higdon (1998). Both interpretations view the SW method as a data augmentation method (Tanner and Wong 1987). More details are given in Section 2.2.

In this paper, we take a third route by interpreting SW as a Metropolis-Hastings step using the auxiliary variables for proposing the moves. Each step is a reversible jump (Green 1995), observing 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 as follows. We start with a background introduction on the Potts model and the Swendsen-Wang algorithm in Section (2). Then we present our generalized method through the Metropolis-Hastings perspective in Section (3). Section (4) shows the first application to image segmentation. Then we proceed to the multi-grid and multi-level cluster algorithm in Section (5). The motion experiments are reported in Section (6). We will compare the performance of our method with the single site Gibbs sampler. The paper is concluded in Section (7) with discussions.

2 Background: Potts model, SW and interpretations

In this section, we review the Potts model, the SW method and its two interpretations.

2.1 SW for Potts models

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

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

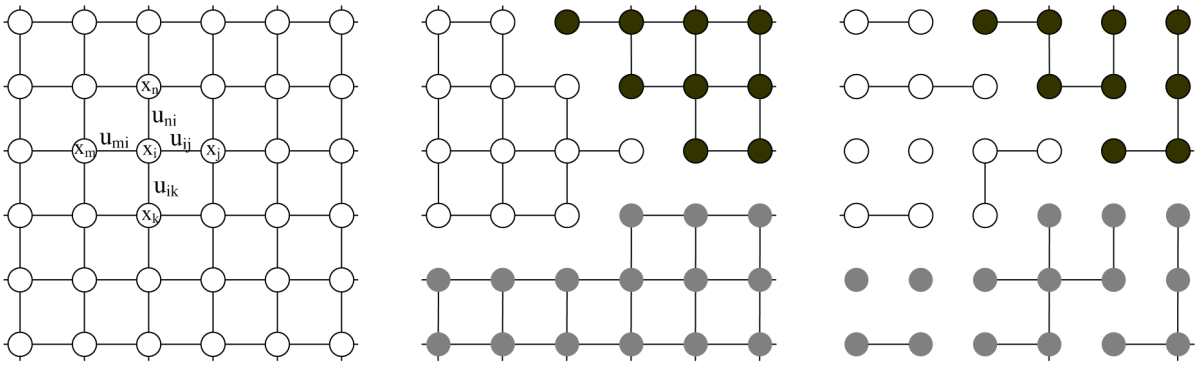
where $\mathbf{1}(x_i = x_j)$ is the indicator function, equal to 1 if condition $x_i = x_j$ is observed, and 0 otherwise. In more general cases, $\beta = \beta_{ij}$ may be position dependent. We usually consider the ferro-magnetic system having $\beta > 0$, which favors the same color for neighboring vertices. The Potts model is used as a prior probability in many Bayesian inference tasks. As Fig.1.(a) illustrates, the SW method introduces a set of auxiliary variables on edges,

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

The edge $< i, j >$ is turned off if and only if $\mu_{ij} = 0$. The μ_{ij} are conditionally independent given $\mathbf{X}$ and each μ_{ij} follows a Bernoulli distribution conditional on x_i, x_j.

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

Thus $\mu_{ij} = 1$ with probability ρ if $x_i = x_j$, and $\mu_{ij} = 1$ with probability 0 if $x_i \neq x_j$.



(a) G_0 (b) G (c) CCP

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

The SW method iterates two steps.

1. **The clustering step**. Given the current state $\mathbf{X}$, the auxiliary variables in $\mathbf{U}$ are sampled according to eqn. (3). First, all edges $< i, j >$ with $x_i \neq x_j$ are turned off deterministically, as Fig. 1.(b) shows. Then, each of the remaining edges is turned off with probability $1 - \rho$. The edges E are divided into the "on" and "off" sets respectively depending on whether $\mu_{ij} = 1$ 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 $\{\text{cp}_1, ..., \text{cp}_K\}$, shown in Fig. 1.(c). We denote the set of connected components by

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

All vertices in a connected component cp_i have the same color.

2. **The swapping step**. One connected component $\text{cp} \in \text{CP}(\mathbf{U})$ is selected at random and a color y is assigned to all vertices in cp. The color 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 swapping for all the connected components in CP($\mathbf{U}$) independently.

In a modified version due to Wolff (1989), a vertex $v \in V$ is chosen and a connected component is grown following the Bernoulli trials on edges around v. This saves computation in the clustering step, but larger components have a higher chance to be selected.

2.2 SW Interpretations and generalizations

There have been a number of different interpretations and generalizations of the Swendsen-Wang method, all viewing it as a data augmentation method (Tanner and Wong 1987).

The first interpretation (Edward and Sokal 1988) on the Potts model makes a connection with the Random Cluster Model. The variables $\mathbf{X}$ and $\mathbf{U}$ are viewed as jointly being sampled from

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

The marginal probability $p_{\text{ES}}(\mathbf{X})$ is the Potts model, while the marginal $p_{\text{ES}}(\mathbf{U})$ is the Random Cluster Model. The Swendsen-Wang algorithm is viewed as sampling from the joint probability $p_{ES}(\mathbf{X}, \mathbf{U})$ by alternatively sampling the two conditional probabilities $p_{\text{ES}}(\mathbf{U}|\mathbf{X})$ and $p_{\text{ES}}(\mathbf{X}|\mathbf{U})$. This interpretation lead to the design of the bounding chain (Huber 2002) for exact sampling.

Of particular interest is the second interpretation, due to Higdon (1998). This method works in a more general setup where the probability considered takes the form

$$\pi(\mathbf{X}) \propto p_0(\mathbf{X}) \prod_{k\in K} b_k(\mathbf{X}), \tag{8}$$

For the Potts model, we have $b_k(\mathbf{X}) = \psi(x_i, x_j) \propto e^{\beta \mathbf{1}(x_i = x_j)}$.

The data is augmented with a set $\mathbf{U} = \{u_k, k \in K\}$ of continuous auxiliary variables $u_k \sim U([0, b_k(\mathbf{X})])$. Then $\mathbf{X}$ and $\mathbf{U}$ are being jointly sampled from

$$p_{\mathrm{HGD}}(\mathbf{X},\mathbf{U}) \propto p_0(\mathbf{X}) \prod_{k\in K} \mathbf{1}[0 \le u_k \le b_k(\mathbf{X})]. \quad (9)$$

Again, the marginal distribution $p_{\mathrm{HGD}}(\mathbf{X})$ is the posterior probability (8) and the algorithm proceeds by alternatively sampling from the two conditional probabilities $p_{\mathrm{HGD}}(\mathbf{U}|\mathbf{X})$ and $p_{\mathrm{HGD}}(\mathbf{X}|\mathbf{U})$. We have

$$p_{\mathrm{HGD}}(\mathbf{X}|\mathbf{U}) \propto p_0(\mathbf{X}) \prod_{k\in K} \mathbf{1}[u_k \le b_k(\mathbf{X})], \quad (10)$$

which means that $\mathbf{X}|\mathbf{U}$ is distributed according to $p_0(\mathbf{X})$ with the constraints $u_k \le b_k(\mathbf{X})$.

In the case of the Potts model, the constraints state that in each connected component (cluster) all nodes have the same label. The clusters are observed to be independent. In general however, sampling $\mathbf{X}$ with the constraints is a difficult task (Higdon 1998, p.5, Besag and Green 1993).

Higdon (1998) also introduced a partial decoupling method in which the auxiliary variables are sampled from $U([0, b_k(\mathbf{X})^{\delta_k}])$, which for the Potts model implies that the formed clusters are not independent anymore. Again, in general, obtaining samples from the conditional probability $p_{\mathrm{HGD}}(\mathbf{X}|\mathbf{U})$ is difficult, and Higdon (1998) presents only applications with Markov Random Fields based on pairwise priors.

Our method generalizes the partial decoupling in three ways. First, it does not assume any particular (factorized) form of the probability distribution. Second, it gives an explicit acceptance probability, which greatly increases the method's applicability. Third, we present applications using higher level priors that are not based on pairwise potentials.

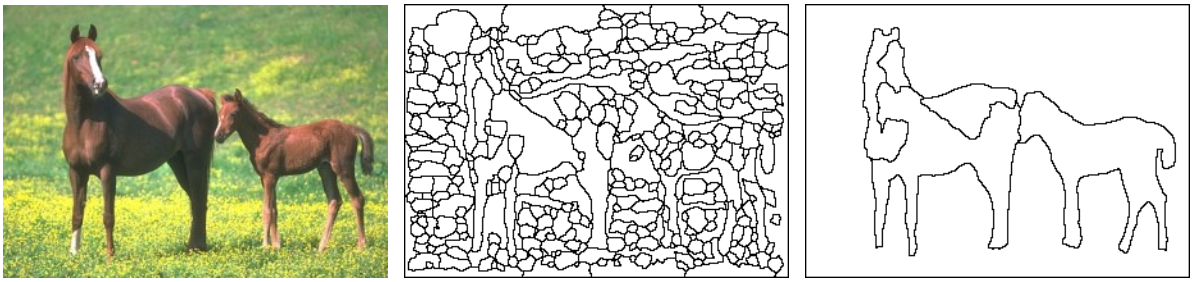
In the next section, we will describe our simple and explicit method based on Metropolis-Hastings, that uses auxiliary variables and samples from an arbitrary probability.

3 Generalizing SW to arbitrary probabilities on graphs

In this section, we generalize the SW to arbitrary probabilities from the perspective of the Metropolis-Hastings method (Metropolis et al 1953, Hastings 1970). Our method iterates three steps: (i) a clustering step driven by data, (ii) a label swapping step which can introduce new labels, and (iii) an acceptance step for the proposed labeling. Worth mentioning

is simplicity of the formula for calculating the acceptance probability.

We describe the three steps in the next three subsections, and then we show how our method reduces to the original SW when working on the Potts models. Many of the results in this section have been derived in Barbu and Zhu (2005) and have been mentioned here just for the sake of completeness. For more details and proofs, the reader is refered to Barbu and Zhu (2005).



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

Figure 2: Example of image segmentation. (a) Input image. (b) Atomic regions by edge detection followed by edge tracing and contour closing, each being a vertex of the graph $\mathbf{G}$. (c) Segmentation (labeling) result where each atomic region is assigned a color or label.

We illustrate the algorithm by an image segmentation example, shown in Fig. 2. The input image $\mathbf{I}$, shown in Fig. 2.(a), is decomposed into a number of "atomic regions" (to reduce the graph size), as seen in Fig. 2.(b). They are obtained by edge detection, edge tracing and contour closing in a preprocessing stage. Each atomic region has nearly constant intensity and is a vertex of the graph $\mathbf{G}$. Two vertices are connected if their atomic regions are adjacent (i.e. share a common boundary). Fig. 2.(c) is a result of our algorithm, optimizing a Bayesian probability $\pi(\mathbf{X}) = \pi(\mathbf{X}|\mathbf{I})$ (see section 4 for details), where $\mathbf{X}$ is a labeling (coloring) of the vertices of $\mathbf{G}$. Note that the number of labels L is unknown, and we do not distinguish between 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 $\mathbf{U} = \{\mu_{ij} :< i, j >\in E\}$ on the edges, 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,

$$\mu_{ij}|(x_i, x_j) \sim \text{Bernoulli}(q_{ij}\mathbf{1}(x_i = x_j)), \quad \forall < i, j > \in E. \tag{11}$$

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

For the image segmentation example, q_{ij} is an empirical measure of the similarity between the intensity statistics at v_i and v_j (see Section 4 for details).

The design of q_{ij} is application specific and is part of the so called discriminative methods. In Sections 4 and 6 we will show how to define the edge weights for our particular applications. Our method will work with any q_{ij}, but a good choice will inform the clustering step and achieve faster convergence.

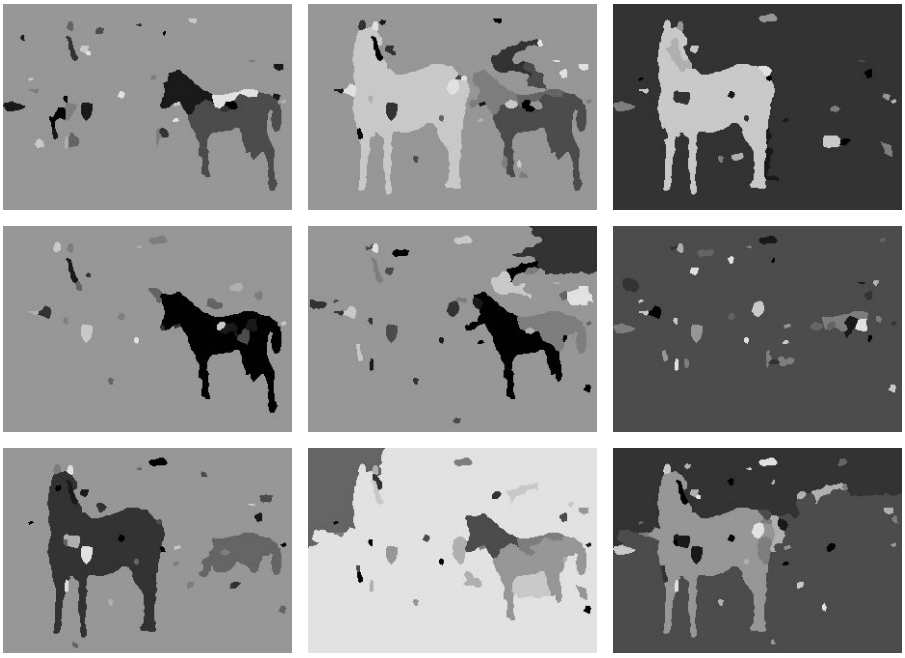


Figure 3: Nine examples of the connected components for the horse image computed using discriminative edge probabilities given that $\mathbf{X}$ is a uniform color $\mathbf{X} = c$ for all vertices.

Fig. 3 shows nine clustering samples of the horse image. For each example, we initialize the graph by setting all vertices to the same color ($\mathbf{X} = c$) and then sample the edge probabilities independently,

$$\mathbf{U}|\mathbf{X} = c \sim \prod_{<i,j>\in E} \text{Bernoulli}(q_{ij}). \tag{12}$$

The connected components in CP($\mathbf{U}$) are shown in different gray levels. For Fig. 3, this procedure is repeated nine times, each time starting from the same state $\mathbf{X} = c$, to show different realizations of the clustering step. 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 probabilities.

3.2 Step 2: swapping of colors

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

$$\text{CP}(\mathbf{U}|\mathbf{X}) = \{\text{cp}_i : i = 1, 2, ..., N(\mathbf{U}|\mathbf{X})\}. \tag{13}$$

One connected component $R \in \text{CP}(\mathbf{U}|\mathbf{X})$ is selected at random by sampling from a probability based only on $\mathbf{U}$, usually uniform over the set of connected components, or proportional to $|R|$. Let the current color of R be $\mathbf{X}_R = \ell \in \{1, 2, ..., n\}$, and select a new color $\ell' \in \{1, 2, ..., n, n+1\}$ for R with probability $q(l'|R, \mathbf{X})$ (to be designed) independent of $\mathbf{U}$, obtaining a new state $\mathbf{X}'$. There are three cases, shown in Fig. 4.

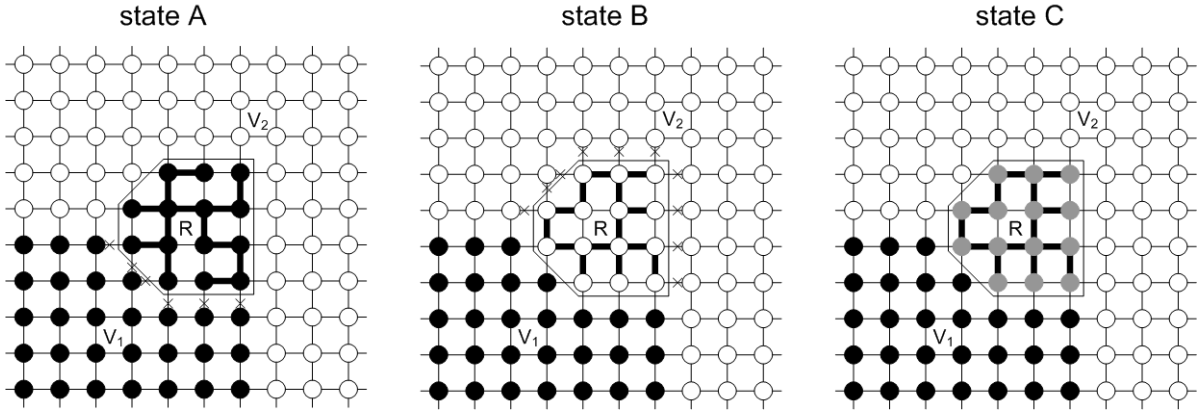


Figure 4: Three labeling states $\mathbf{X}_A, \mathbf{X}_B, \mathbf{X}_C$ which differ only in the color of a cluster R.

1. The canonical case: R is a proper subset of V_ℓ 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 n in $\mathbf{X}'$. The moves between $\mathbf{X}_A \leftrightarrow \mathbf{X}_B$ in Fig. 4 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. 4 are examples.

3. The split case: R is a proper subset of V_ℓ and $\ell' = n+1$. V_ℓ is split into two pieces and the number of distinct colors increases to $n+1$ in $\mathbf{X}'$. The moves $\mathbf{X}_A \to \mathbf{X}_C$ and $\mathbf{X}_B \to \mathbf{X}_C$ in Fig. 4 are examples.

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

3.3 Step 3: accepting the swap

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

$$\alpha(\mathbf{X} \to \mathbf{X}') = \min\{1, \frac{q(\mathbf{X}' \to \mathbf{X})}{q(\mathbf{X} \to \mathbf{X}')} \cdot \frac{\pi(\mathbf{X}')}{\pi(\mathbf{X})}\}. \tag{14}$$

where $q(\mathbf{X}' \to \mathbf{X})$ 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 remains at state $\mathbf{X}$.

For the canonical case, there is a unique path for moving between $\mathbf{X}$ 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 swapping steps respectively: (i) the probability ratio for selecting R as candidate in the clustering step in states $\mathbf{X}$ and $\mathbf{X}'$, and (ii) the probability ratio for selecting the new label for R in the swapping step.

$$\frac{q(\mathbf{X}' \to \mathbf{X})}{q(\mathbf{X} \to \mathbf{X}')} = \frac{q(R|\mathbf{X}')}{q(R|\mathbf{X})} \cdot \frac{q(\mathbf{X}_R = \ell | R, \mathbf{X}')}{q(\mathbf{X}_R = \ell' | R, \mathbf{X})}. \tag{15}$$

For the split and merge cases, it can happen that there are two paths between $\mathbf{X}$ and $\mathbf{X}'$. For the merge case, this only happens when $R = V_\ell$ for some ℓ and at the same time $V_{\ell'}$ is a connected component. Then one path is obtained by choosing $R = V_\ell$ and merging it to $V_{\ell'}$ and the other path is obtained by choosing $R = V_{\ell'}$ and merging it with V_ℓ. The

two paths in the split case are the reverse of the merge paths. The conclusion above still holds if the proposal probability $q(\mathbf{X}_R = \ell | R, \mathbf{X}')$ satisfies the following condition:

$$\frac{q(\mathbf{X}_R = \ell' | R = V_\ell, \mathbf{X}')}{q(\mathbf{X}_R = \ell | R = V_{\ell'}, \mathbf{X})} = \frac{q(\mathbf{X}_R = \ell | R = V_{\ell'}, \mathbf{X}')}{q(\mathbf{X}_R = \ell' | R = V_\ell, \mathbf{X})} \tag{16}$$

which means that the ratio of the proposal probabilities for split and for merge on each path is the same.

One commonly used proposal probability satisfying the above condition is:

$$q(l | R, \mathbf{X}) \propto \begin{cases} a & \text{if } R \text{ adjacent to } V_l \\ b & \text{if } R \text{ not adjacent to } V_l \\ c & \text{if } l \notin L \text{ (new region)} \end{cases} \tag{17}$$

where the values a, b, c are positive arbitrary constants (e.g. $a = 10, b = 1, c = 0.1$).

To obtain an explicit formula for the acceptance probability, we need the following

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

$$\mathcal{C}(R, V_k) = \{< i, j >: \ i \in R, j \in V_k \backslash R\}, \ \ \forall k. \tag{18}$$

The crosses in Fig.4.(a) and (b) show the cut $\mathcal{C}(R, V_1)$ and $\mathcal{C}(R, V_2)$ respectively. In Fig.4.(c), $R = V_3$ and thus $\mathcal{C}(R, V_3) = \emptyset$ and $\prod_{<i,j> \in \mathcal{C}(R,V_3)} (1 - q_{ij}) = 1$.

A key observation by Barbu and Zhu (2005) is that the probability ratio $\frac{q(R|\mathbf{X}')}{q(R|\mathbf{X})}$ for proposing R only depends on the cuts between R and the rest of the vertices,

$$\frac{q(R|\mathbf{X})}{q(R|\mathbf{X}')} = \frac{\prod_{<i,j> \in \mathcal{C}(R,V_\ell)} (1 - q_{ij})}{\prod_{<i,j> \in \mathcal{C}(R,V_{\ell'})} (1 - q_{ij})}. \tag{19}$$

where q_{ij}'s are the edge probabilities.

The acceptance probability is given by the following

Theorem 1 (Barbu and Zhu, 2005) *The acceptance probability for the proposed swapping is*

$$\alpha(\mathbf{X} \to \mathbf{X}') = \min\{1, \frac{\prod_{<i,j> \in \mathcal{C}(R,V_{\ell'})} (1 - q_{ij})}{\prod_{<i,j> \in \mathcal{C}(R,V_\ell)} (1 - q_{ij})} \cdot \frac{q(\mathbf{X}_R = \ell | R, \mathbf{X}')}{q(\mathbf{X}_R = \ell' | R, \mathbf{X})} \cdot \frac{\pi(\mathbf{X}')}{\pi(\mathbf{X})}\}. \tag{20}$$

[Proof] See Theorem 2 in Barbu and Zhu (2005).

3.4 Interpretation of the SW algorithm from the Metropolis-Hastings perspective

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

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

$$\frac{q(R|\mathbf{X})}{q(R|\mathbf{X}')} = \frac{\prod_{<i,j>\in\mathcal{C}(R,V_\ell)}(1-q_{ij})}{\prod_{<i,j>\in\mathcal{C}(R,V_{\ell'})}(1-q_{ij})} = \exp\{\beta(|\mathcal{C}(R,V_{\ell'})| - |\mathcal{C}(R,V_\ell)|)\}, \tag{21}$$

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

As $\mathbf{X}$ and $\mathbf{X}'$ only differ in the label of R, the potentials for the Potts model only differ at the cuts between R and V_ℓ and $V_{\ell'}$ respectively.

Proposition 2 *For the Potts model* $\pi(\mathbf{X}) = p_o(\mathbf{X}) = \pi_{\mathrm{PTS}}(\mathbf{X})$*,*

$$\frac{\pi_{\mathrm{PTS}}(\mathbf{X}_R = \ell'|\mathbf{X}_{\partial R})}{\pi_{\mathrm{PTS}}(\mathbf{X}_R = \ell|\mathbf{X}_{\partial R})} = \exp\{\beta(|\mathcal{C}(R,V_\ell)| - |\mathcal{C}(R,V_{\ell'})|)\} \tag{22}$$

Therefore, following eq. (20) (where the proposal probabilities for the labels are uniform), the acceptance probability for the Potts model is always one, due to cancellation.

$$\alpha(\mathbf{X} \to \mathbf{X}') = 1. \tag{23}$$

Therefore the third acceptance step is always omitted. This interpretation is related to the Wolff (1989) modification (see also Liu 2001, p.157).

3.5 The multiple swapping scheme

Given a set of connected components $\mathrm{CP}(\mathbf{U}|\mathbf{X})$ (see eqn. (13)) after the clustering step, instead of swapping a single component R, we can swap (change) all (or a 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 state $\mathbf{X}'$ after swapping. Let $E_{\mathrm{on}}(\mathbf{X}) \subset E$ and $E_{\mathrm{on}}(\mathbf{X}') \subset E$ be the subsets of edges connecting vertices of same color in $\mathbf{X}$ and $\mathbf{X}'$ respectively. We define two cuts as set differences

$$\mathcal{C}(\mathbf{X} \to \mathbf{X}') = E_{\mathrm{on}}(\mathbf{X}') - E_{\mathrm{on}}(\mathbf{X}), \text{ and } \mathcal{C}(\mathbf{X}' \to \mathbf{X}) = E_{\mathrm{on}}(\mathbf{X}) - E_{\mathrm{on}}(\mathbf{X}'), \tag{24}$$

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

Proposition 3 *The acceptance probability of the multiple swapping scheme is*

$$\alpha(\mathbf{X} \to \mathbf{X}') = \min\{1, \frac{\prod_{<i,j>\in\mathcal{C}(\mathbf{X}\to\mathbf{X}')}(1-q_{ij})}{\prod_{<i,j>\in\mathcal{C}(\mathbf{X}'\to\mathbf{X})}(1-q_{ij})} \frac{\prod_{\text{cp}\in D(\mathbf{X},\mathbf{X}')} q(\mathbf{X}'_{\text{cp}}|\text{ cp})}{\prod_{\text{cp}\in D(\mathbf{X},\mathbf{X}')} q(\mathbf{X}_{\text{cp}}|\text{ cp})} \cdot \frac{\pi(\mathbf{X}')}{\pi(\mathbf{X})}\} \tag{25}$$

[Proof] See Barbu (2005).

Observe that when $D = \{R\}$ is a single connected component, this reduces to Thm. 1.

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

$$\mathcal{K}(\mathbf{X}, \mathbf{X}') = q(\mathbf{X} \to \mathbf{X}')\alpha(\mathbf{X} \to \mathbf{X}') > 0, \ \ \forall\, \mathbf{X}, \mathbf{X}' \in \Omega. \tag{26}$$

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

4 Experiment 1: image segmentation

Our first experiment tests the cluster algorithm in an image segmentation task. The objective is to partition the image into a number of disjoint regions (as shown in Figs. 2 and 3) so that each region is coherent in the sense of fitting to some image models. The final result should optimize a Bayesian posterior probability $\pi(\mathbf{X}) \propto \mathcal{L}(\mathbf{I}|\mathbf{X})p_o(\mathbf{X})$ with likelihood $\mathcal{L}(\mathbf{I}|\mathbf{X})$ and prior $p_o(\mathbf{X})$. In such a problem, $\mathbf{G}$ is an adjacency graph whose vertices V are a set of atomic regions (see Figs. 2 and 3). Usually $|V|$ is in the order of hundreds. The edge probability should represent a good similarity measure between the intensity models of the atomic regions. As an approximate model for each atomic region $v \in V$, we choose a 15-bin intensity histogram h normalized to 1. We choose the edge probability as

$$q_{ij} = \exp\{-\frac{1}{2}(KL(h_i||\, h_j) + KL(h_j||\, h_i))\}, \tag{27}$$

where $KL()$ is the Kullback-Leibler divergence between the two histograms. In our experiments we observed that this edge probability is a good similarity measure and leads to good clustering, as Fig. 3 shows.

Now we briefly define the target probability in this experiment. Let $\mathbf{X} = (V_1, ..., V_{\mathrm{L}})$ be a coloring with L labels of the graph, L being an unknown variable. Each set V_k has a model Θ_k with parameters θ_k. Different colors are assumed to be independent. Therefore we have,

$$\pi(\mathbf{X}) = \pi(\mathbf{X}|\mathbf{I}) \propto \prod_{k=1}^{\mathrm{L}} [\mathcal{L}(\mathbf{I}(V_k); \theta_k) p_o(\Theta_k)] p_o(\mathbf{X}). \tag{28}$$

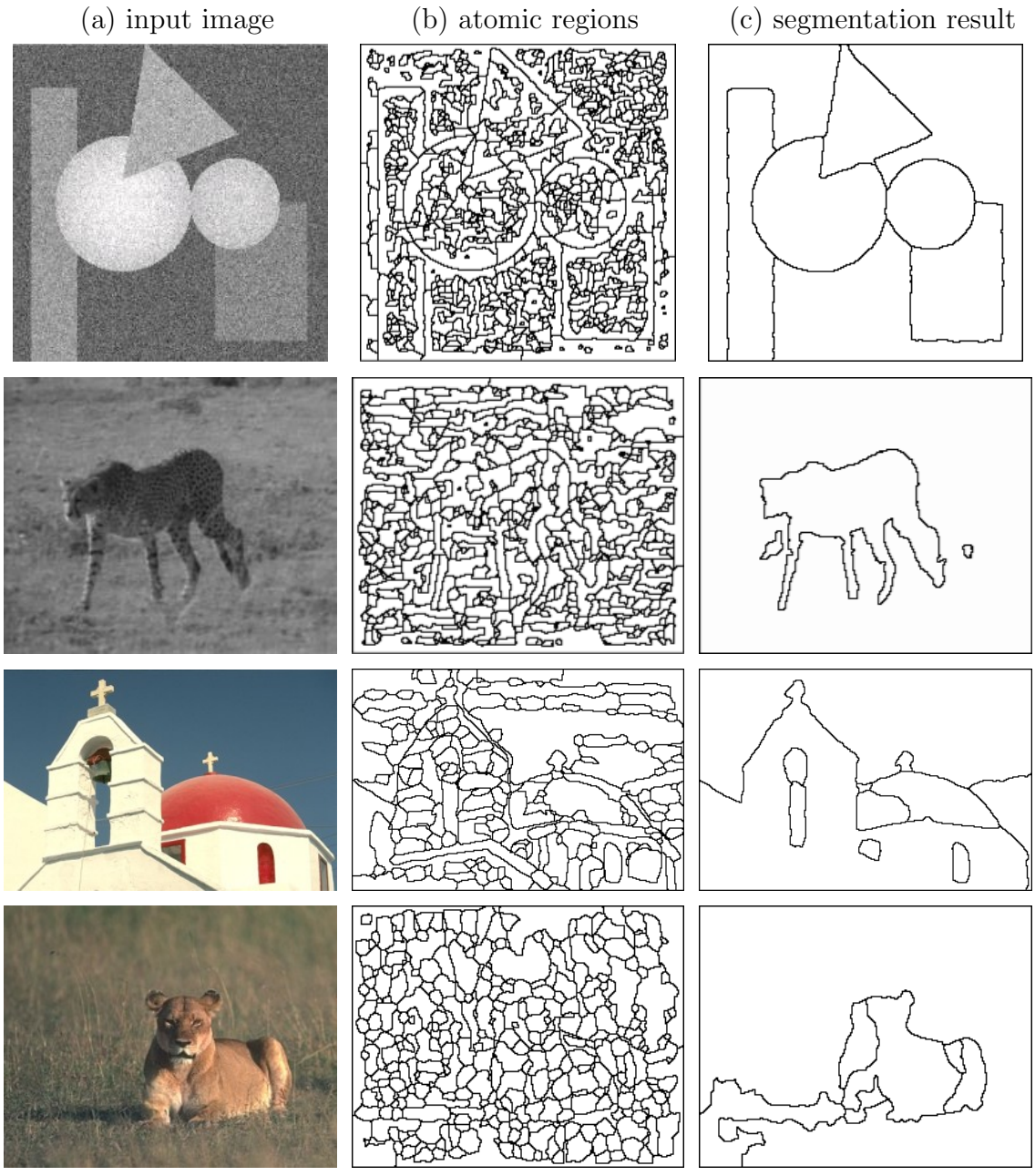


Figure 5: More results for image segmentation.

We selected three types of simple intensity models to account for different image prop-

erties. 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, such as vegetation,

$$\mathbf{I}(x, y;\, \theta_0) \sim \mathcal{H} \text{ iid},\ \forall (x, y) \in V_k, \text{where } \theta_0 = \mathcal{H}. \tag{29}$$

The other two are regression models for the smooth change of intensities in the two-dimensional image plane (x, y), with the residues following a histogram $\mathcal{H}$

$$\mathbf{I}(x, y;\, \theta_1) = \beta_0 + \beta_1 x + \beta_2 y + \mathcal{H} \text{ iid},\ \forall (x, y) \in V_k. \tag{30}$$

$$\mathbf{I}(x, y;\, \theta_2) = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 xy + \beta_5 y^2 + \mathcal{H} \text{ iid},\ \forall (x, y) \in V_k. \tag{31}$$

where $\theta_1 = (\beta_0, \beta_1, \beta_2, \mathcal{H})$ and $\theta_2 = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \mathcal{H})$.

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

$$\mathcal{L}(\mathbf{I}(V_k); \theta_k) \propto \prod_{v \in V_k} \mathcal{H}(\mathbf{I}_v) = \prod_{j=1}^{B} \mathcal{H}_j^{n_j} = \exp(-|V_k| \text{entropy}(\mathcal{H})). \tag{32}$$

The model complexity is penalized by a prior probability $p_o(\Theta_k)$, while 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 specialized model fitting steps. 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 a 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

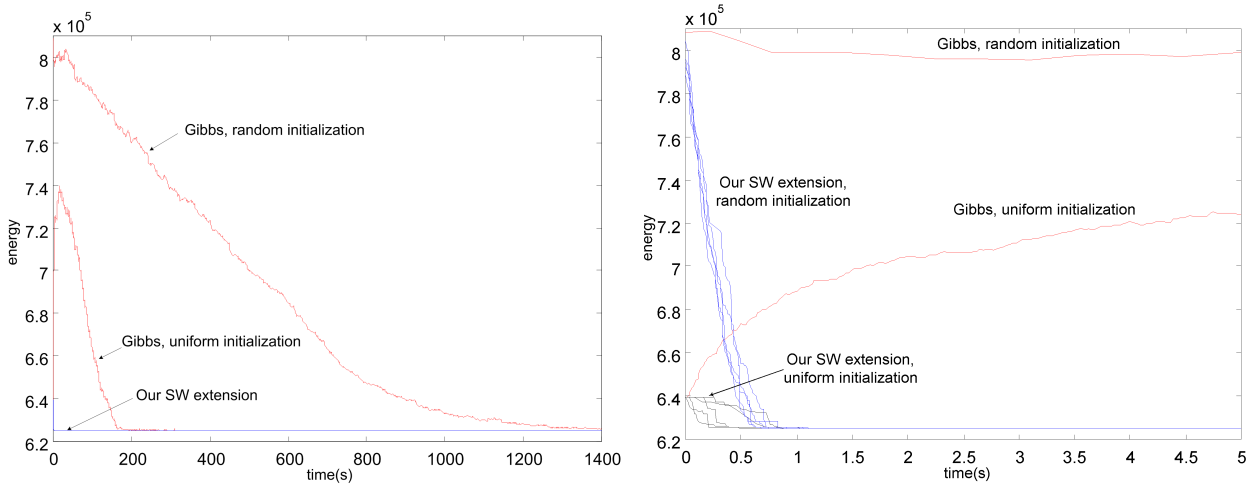
$$p_o(\mathbf{X}) \propto \exp\{-\alpha_0 \mathrm{L} - \alpha_1 m - \alpha_2 \sum_{k=1}^{m} \text{Area}(r_k)^{0.9}\}. \tag{33}$$

Experimentally we choose $\alpha_0 = 35, \alpha_1 = 15$ for all experiments. The scale factor α_2 controls the size of the segmentation regions and takes values between 5 and 15 and is chosen for each example by experiment.

The reassignment probability $q(\ell | R, \mathbf{X})$ is defined in terms of the intensity histogram $H(R)$ of the region R to be reassigned and the histogram $H(V_\ell)$ of each region V_ℓ.

$$q(\ell | R, \mathbf{X}) \propto \begin{cases} 10 e^{-KL(H(R), H(V_\ell))} & \text{if } R \text{ adjacent to } V_\ell \\ e^{-KL(H(R), H(V_\ell))} & \text{if } R \text{ not adjacent to } V_\ell \\ 0.1 & \text{if } \ell \notin L \text{ (new region)} \end{cases} \tag{34}$$

For the image segmentation example (horse) shown in Figs. 2 and 3, we compare the cluster method with the single-site Gibbs sampler and the results are displayed in Fig. 6. Since our goal is to maximize the posterior probability $\pi(\mathbf{X})$, we add an annealing scheme with a high initial temperature T_o that decreases to a low temperature (0.5 in our experiments). We plot $-\ln \pi(\mathbf{X})$ vs. CPU time in seconds on a Pentium IV PC. For the Gibbs sampler, we needed to start with a high initial temperature ($T_o \sim 100$)) and use a slow annealing schedule ($5000|V|$ steps, $|V|$ being the graph size) to reach the same energy level as our algorithm. We experimentally observed that starting the Gibbs sampler with a lower initial temperature or using a faster annealing schedule will cause it to remain stuck in a local minimum of higher energy, and not reach the final energy in any reasonable time. The cluster algorithm can run at low temperature, starting with an initial temperature $T_o = 15$ and using a fast annealing scheme ($15|V|$ steps). Fig. 6.(a) plots the two algorithms at the first $1,400$ seconds, and Fig. 6.(b) is a zoomed-in view of the first 5 seconds.

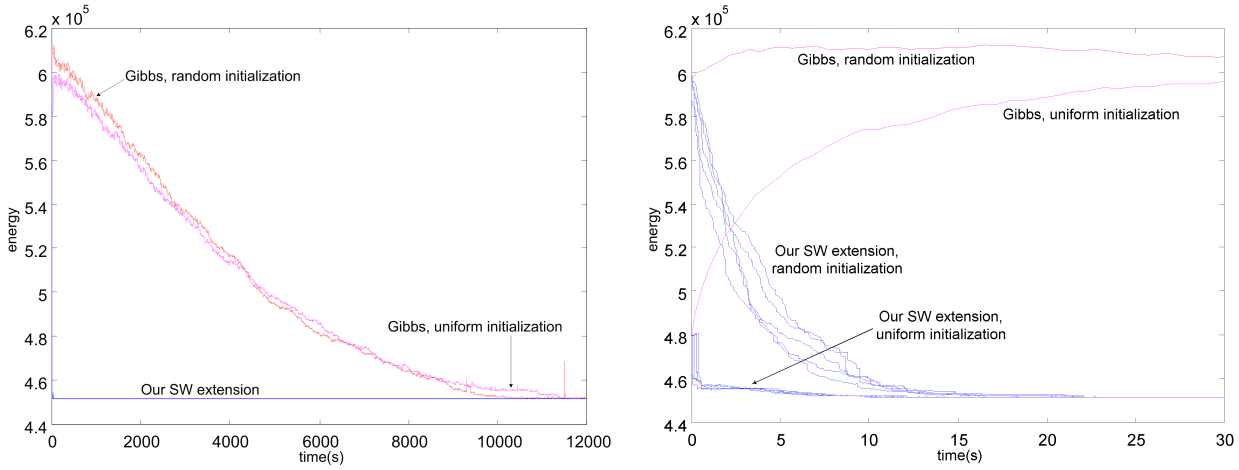


(a) convergence CPU time in seconds (b) Zoomed-in view of the first 5 seconds.

Figure 6: The plot of $-\ln \pi(X)$ vs. CPU time for both the Gibbs sampler and our algorithm for the horse image. The algorithms are compared by measuring the CPU time in seconds on a Pentium IV PC. (a). Plot of 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.

Each algorithms was run with two initializations. One is a random labeling of the atomic regions and thus has higher energy, while the other initialization sets all vertices to

the same color. The cluster algorithm is run five times in each case. All runs converged to one solution (see Fig.2.(c)) within 1 second, which is hundreds of times faster than the Gibbs sampler.



(a) (b)

Figure 7: 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.5. (a). The first 12,000 seconds. (Right) Zoomed-in view of the first 30 seconds. Our algorithm is 1,000 times faster in this case.

Fig.5 shows four more experimental results. Using the same method as in the horse image, we plot $-\ln \pi(\mathbf{X})$ against CPU time in Fig. 7 for the first image of Fig.5. In experiments, we also compared the effect of the edge probabilities. The clustering algorithm is hundreds of times times slower if we use a constant edge probability $\mu_{ij} = c \in (0, 1)$ as the original SW method does. The single-site Gibbs sampler is an example with $q_{ij} = 0$, $\forall\, i, j$. The reader is also referred to Barbu and Zhu (2005) for more results and an evaluation of the influence of graph edge weights on the convergence speed.

5 The Multi-level and Multi-grid cluster algorithms

When the graph $\mathbf{G}$ is large, for example, $|V| \sim 10^4 - 10^6$ in image analysis, the clustering step has to sample many edges and is computationally costly. This section presents two strategies for improving the speed – the multi-grid and multi-level cluster algorithm. 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)

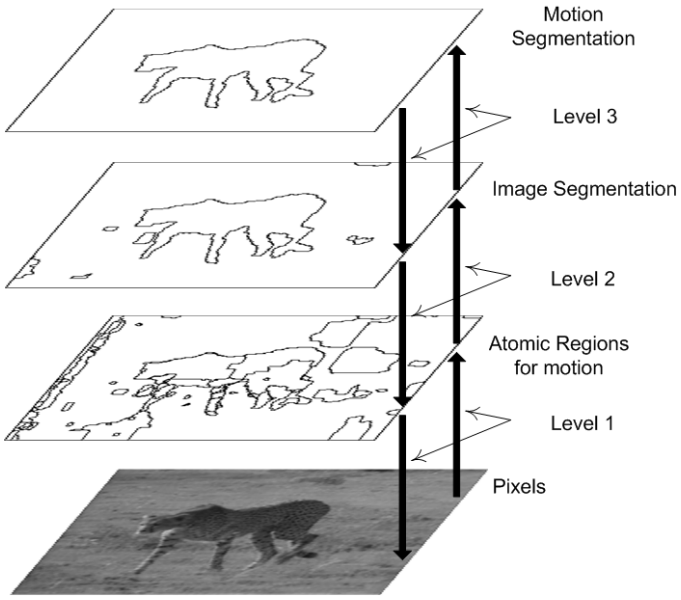


Figure 8: The cluster algorithm on multiple graph levels for motion segmentation. Connected components are frozen and collapsed into single vertices in the level above.

5.1 Rationale for the multi-level and multi-grid cluster algorithms

The multi-level cluster algorithm is motivated by the problem of hierarchic graph labeling. Fig. 8 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 planar velocity (i.e. optical flow) of the pixels in the second frame based on their displacement between the two frames, (ii) segment (group) the pixels into regions of coherent intensities and motion, and (iii) further group the intensity regions into moving objects, such as the running cheetah and the grass background where each object should have coherent motion in the image plane.

This situation can be represented as a three-level labeling $\mathbf{X} = (\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)})$, of three graphs shown in Fig. 8,

$$\{\mathbf{G}^{(s)} =< V^{(s)}, E^{(s)} > :\ s = 0, 1, 2\}. \tag{35}$$

$\mathbf{G}^{(0)}$ is the image lattice with each vertex being a pixel. The pixels are labeled by $\mathbf{X}^{(0)}$ according to their intensity and planar motion velocity. The range of possible velocities

is discretized and the pixel label encodes the value of the pixel's velocity. By the label $\mathbf{X}^{(0)}$, the pixels are grouped into a number of small regions of nearly constant intensity and 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 vertex size is reduced from $\sim 10^5$ in $\mathbf{G}^{(0)}$ to $\sim 10^2$ in $\mathbf{G}^{(1)}$ and to ~ 10 in $\mathbf{G}^{(2)}$.

In multi-grid sampling, we introduce an "attention window" Λ (see Fig. 9) which may change location and size over time. To save computation, the cluster algorithm is limited to the attention window at each step, and this is equivalent to sampling a conditional probability, $\mathbf{X}_\Lambda \sim \pi(\mathbf{X}_\Lambda|\mathbf{X}_{\bar{\Lambda}})$.

The multi-grid and multi-level methods in the next two subsections are ways for designing sub-kernels that are reversible.

5.2 Multi-grid clustering

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

$$V = V_\Lambda \cup V_{\bar{\Lambda}}, \text{ and } \mathbf{X} = (\mathbf{X}_\Lambda, \mathbf{X}_{\bar{\lambda}}). \tag{36}$$

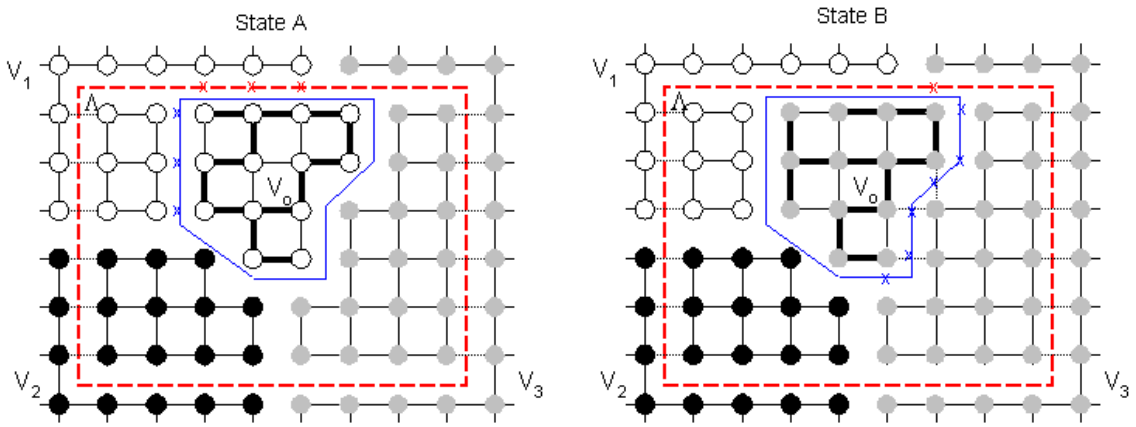


Figure 9: Multigrid swapping: computation is restricted to different "attention" windows Λ of various sizes, with the rest of the labels fixed.

For example, Fig. 9 displays a rectangular window Λ (in red dashed) in a lattice $\mathbf{G}$. The window Λ cuts some edges in each subset $V_k, k = 1, 2, ..., \mathrm{L}$, which we denote 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. 9 the window Λ intersects with three subsets V_1 (white), V_2 (black), and V_3 (gray), and all edges crossing the (red) rectangle window are cut.

The multi-grid cluster algorithm

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

Proposition 4 *The acceptance probability for changing the label of the candidate cluster R from ℓ in state $\mathbf{X}$ to ℓ' in state $\mathbf{X}'$ within the window Λ is*

$$\alpha(\mathbf{X} \to \mathbf{X}') = \min\{1, \frac{\prod_{<i,j>\in\mathcal{C}(R,V_{\ell'})-\mathcal{C}(V_{\ell'},\Lambda)} (1-q_{ij})}{\prod_{<i,j>\in\mathcal{C}(R,V_{\ell})-\mathcal{C}(V_{\ell},\Lambda)} (1-q_{ij})} \cdot \frac{q(\mathbf{X}_R = \ell|R, \mathbf{X}')}{q(\mathbf{X}_R = \ell'|R, \mathbf{X})} \cdot \frac{\pi(\mathbf{X}')}{\pi(\mathbf{X})}\}. \tag{37}$$

In Fig. 9, we have $\mathbf{X} = \mathbf{X}_A$ and $\mathbf{X}' = \mathbf{X}_B$ ($\ell = 1, \ell' = 3$).

The difference between this equation and equation (20) is that some edges in $\mathcal{C}(V_\ell, \Lambda) \cup \mathcal{C}(V_{\ell'}, \Lambda)$ no longer participate in the computation.

Proposition 5 *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,*

$$\pi(\mathbf{X}_\Lambda|\mathbf{X}_{\bar{\Lambda}})\mathcal{K}(\mathbf{X}_\Lambda, \mathbf{Y}_\Lambda) = \pi(\mathbf{Y}_\Lambda|\mathbf{X}_{\bar{\Lambda}})\mathcal{K}(\mathbf{Y}_\Lambda, \mathbf{X}_\Lambda). \tag{38}$$

It is easy to prove that the multi-grid method is also invariant to the full posterior probability. The proof relies of the fact that if $p(x, y)$ is a two dimensional probability, and $\mathcal{K}$ is a Markov kernel reversible with respect to the conditional probability $p(x|y)$

$$p(x|y)\mathcal{K}(x, x') = p(x'|y)\mathcal{K}_1(x', x), \ \forall x, x'. \tag{39}$$

then $\mathcal{K}$ observes the detailed balance equation after augmenting y

$$p(x, y)\mathcal{K}((x, y), (x', y')) = p(x', y')\mathcal{K}((x', y'), (x, y)). \tag{40}$$

In practice, the attention window Λ is randomly chosen in such a way that its position probability is uniform and its size probability is inversely proportional to the size.

5.3 The multi-level cluster algorithm

Following the notations in Section (5.1), the problem is hierarchical 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 a partition of the lattice

$$\mathbf{X}^{(s)} = \{V_1^{(s)}, V_2^{(s)}, ..., V_{m^{(s)}}^{(s)}\}, \ s = 0, 1, 2. \tag{41}$$

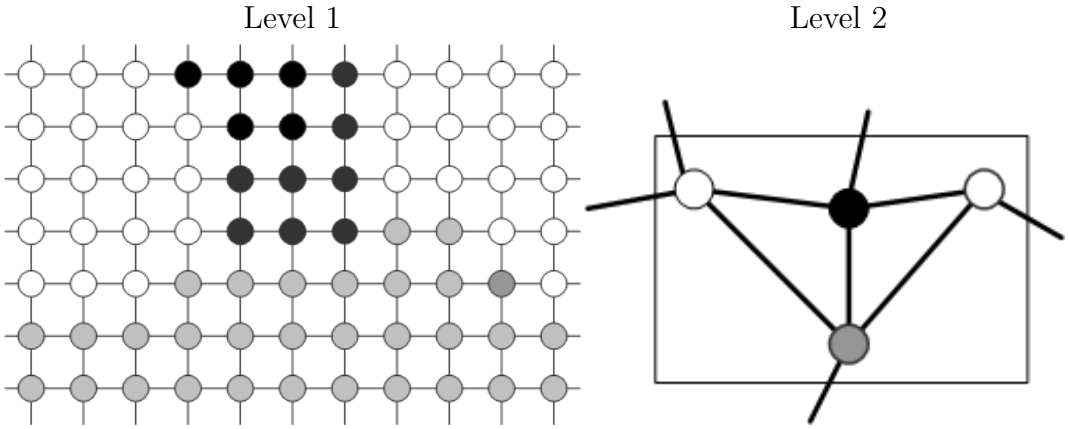


Figure 10: The multi-level cluster algorithm. Computation is performed at different levels of granularity, where the connected components from the lower level collapse into vertices in the higher level.

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

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

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

The multi-level cluster algorithm

1. Select a level s, usually in an increasing order.
2. Cluster the vertices in $\mathbf{G}^{(s)}$ and select a connected component R.
3. Swap the label of R.
4. Accept the swap with probability (20), using the lower levels, denoted by $\mathbf{X}^{(<s)}$, as boundary conditions.

Proposition 6 *Let $\pi(\mathbf{X})$ be a probability on the nested labeling $\mathbf{X} = (\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)})$, and let the kernels of the cluster algorithm on the three levels of graphs be $\mathcal{K}^{(0)}$, $\mathcal{K}^{(1)}$ and $\mathcal{K}^{(2)}$ respectively. Then they observe the detailed balance equations with respect to the conditional probabilities,*

$$\pi(\mathbf{X}^{(s)}|\mathbf{X}^{(<s)})\mathcal{K}^{(s)}(\mathbf{X}^{(s)}, \mathbf{Y}^{(s)}) = \pi(\mathbf{Y}^{(s)}|\mathbf{X}^{(<s)})\mathcal{K}^{(s)}(\mathbf{Y}^{(s)}, \mathbf{X}^{(s)}),\ s = 0, 1, 2. \tag{42}$$

where $\mathbf{X}^{(<s)} = (\mathbf{X}^0, ..., \mathbf{X}^{s-1})$. Therefore the multi-level cluster algorithm is reversible.

6 Experiment 2: hierarchical motion segmentation

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

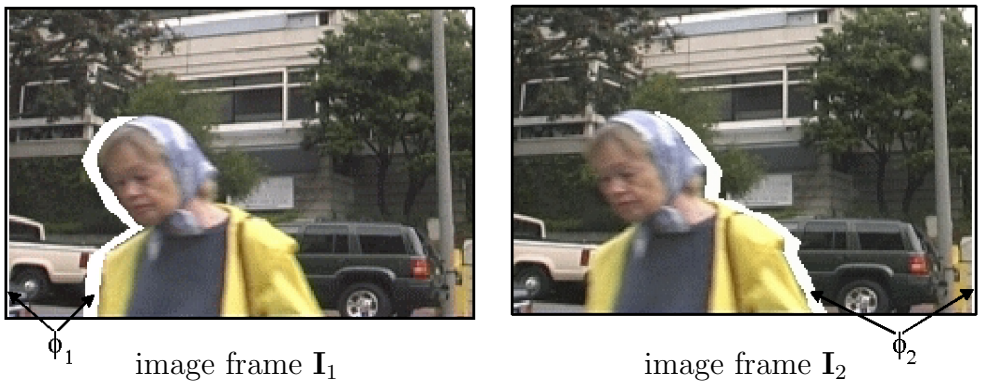


image frame $\mathbf{I}_1$ image frame $\mathbf{I}_2$

Figure 11: Two consecutive image frames in which the foreground and background are moving. The pixels in area ϕ_1 are not seen in $\mathbf{I}_2$ and similarly the pixels in ϕ_2 are not seen in image $\mathbf{I}_1$, and they are called "half-occluded" pixels. The rest of the pixels can be mapped between the two frames.

Let $\mathbf{I}_1, \mathbf{I}_2$ be two consecutive image frames in a video as Fig. 11 illustrates. Due to occlusion, some points are visible in only one image, e.g. the white areas ϕ_1 in $\mathbf{I}_1$ and ϕ_2 in $\mathbf{I}_2$, and are called "half-occluded" points. All other points can be mapped between the two image frames $\mathbf{I}_1, \mathbf{I}_2$. The mapping function is called the "optical flow" field,

$$(u, v) : \mathbf{I}_2 \backslash \phi_2 \mapsto \mathbf{I}_1 \backslash \phi_1. \tag{43}$$

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

$$\mathbf{I}_2(x, y) = \mathbf{I}_1(x - u(x, y), y - v(x, y)) + \mathbf{n}(x, y), \ \ \forall (x, y) \in \mathbf{I}_2 \backslash \phi_2. \tag{44}$$

We discretize the images $\mathbf{I}_1$ and $\mathbf{I}_2$ into lattices Λ_1 and Λ_2 respectively. In our motion analysis problem, we consider the pixels in the second image frame as the lattice $\mathbf{G}^{(0)} = \Lambda_2$, and each pixel has three labels $x = (x^{(0)}, x^{(1)}, x^{(2)})$.

1. The velocity $x^{(0)} = (u, v)$ is discretized into $21 \times 9 = 189$ different planar velocities, since we assume the range of displacements between two consecutive frames to be $-5 \leq u \leq 5, -2 \leq v \leq 2$ with $1/2$ pixel precision. For pixels that have no correspondent in the first frame, i.e. pixels in ϕ_2, their velocities cannot be decided and are denoted by nil. They are labeled based on the context information on their intensity through image segmentation. By this convention, we can consider $x^{(0)}$ as a velocity label $x^{(0)} \in \{\text{nil}, 1, 2, ..., 189\}$.

2. The 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 from section 4.

3. The 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 motion.

To fix notation, we divide the image frames into two parts, namely occluded and non-occluded, $\mathbf{I}_1 = (\mathbf{I}_{1,\phi_1}, \mathbf{I}_{1,\bar{\phi}_1}),\ \mathbf{I}_2 = (\mathbf{I}_{2,\phi_2}, \mathbf{I}_{2,\bar{\phi}_2})$.

The target probability is the Bayesian posterior,

$$\pi(\mathbf{X}) = \pi(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)} | \mathbf{I}_1, \mathbf{I}_2) \propto \mathcal{L}(\mathbf{I}_{1,\bar{\phi}_1} | \mathbf{I}_{2,\bar{\phi}_2}, \mathbf{X}^{(0)}) \mathcal{L}(\mathbf{I}_2 | \mathbf{X}^{(1)}) \pi_o(\mathbf{X}). \tag{45}$$

The first likelihood is specified by the optical flow model,

$$\mathcal{L}(\mathbf{I}_{1,\bar{\phi}_1}|\mathbf{I}_{2,\bar{\phi}_2}, \mathbf{X}^{(0)}) = \prod_{(x,y)\in\Lambda_2\setminus\phi_2} \frac{1}{\sqrt{2\pi}\sigma_o} \exp\{-\frac{1}{2\sigma_o^2}(\mathbf{I}_2(x,y) - \mathbf{I}_1(x-u(x,y), y-v(x,y)))^2\}. \tag{46}$$

where $\sigma_0^2 = \frac{1}{|\Lambda_2\setminus\phi_2|}\sum_{(x,y)\in\Lambda_2\setminus\phi_2}(\mathbf{I}_2(x,y) - \mathbf{I}_1(x-u(x,y), y-v(x,y)))^2$.

The second likelihood is the same as the image segmentation likelihood in Section (4). The prior probability assumes piecewise coherent motion. That is, each moving object $o = 1, 2, ..., \mathrm{L}^{(2)}$ has a constant planar velocity $c_o \in \{1, 2, ..., 189\}$ plus a Markov model for the adjacent velocities. Also each object (and region) has a compact boundary prior.

$$\begin{aligned}\pi_o(\mathbf{X}) \quad &\propto \prod_{o=1}^{\mathrm{L}^{(2)}} \exp\{-\alpha \sum_{v, x^{(2)}(v)=o} |x^{(0)}(v) - c_o|^2 - \beta \sum_{v'\in\partial v} |x^{(0)}(v') - x^{(0)}(v)|\} \\ &\prod_{l=1}^{\mathrm{L}^{(1)}} \exp\{-\gamma|\partial V_l^{(1)}|\} \prod_{i=1}^{\mathrm{L}^{(0)}} \exp\{-\delta|\partial V_i^{(0)}|\} \exp\{-\lambda_0 \mathrm{L}^{(0)} - \lambda_1 \mathrm{L}^{(1)} - \lambda_2 \mathrm{L}^{(2)}\}\end{aligned} \tag{47}$$

where we experimentally set $\alpha = 1, \beta = 0.5, \gamma = 1, \delta = 1, \lambda_0 = 10$ while λ_1, λ_2 are scale parameter for the intensity and motion segmentation respectively (usually $\lambda_1 = 10, \lambda_2 = 5$).

Now we define the edge probabilities and the reassignment probabilities at the three levels of the graph.

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

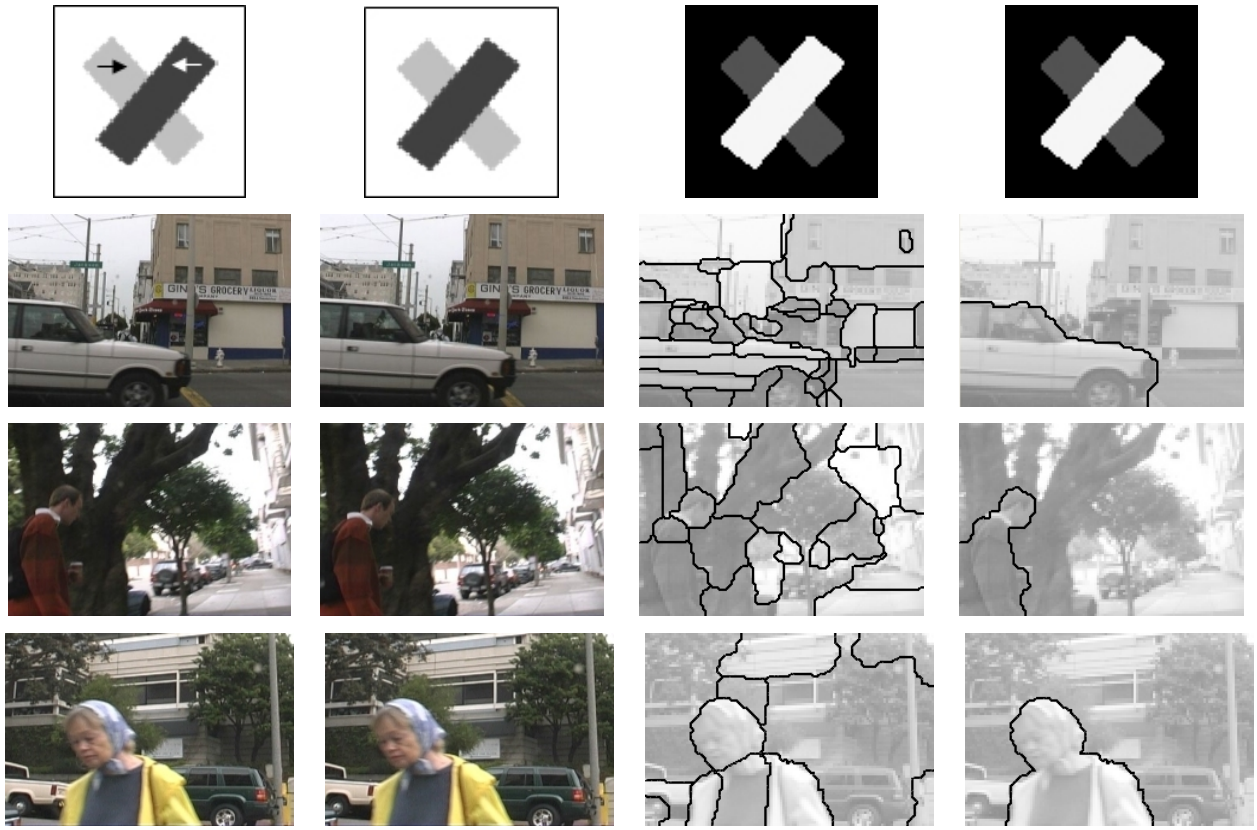
$$\begin{aligned}q^{(0)}(s, s') = \quad &\min_{(u,v)} \exp\{-[|\mathbf{I}_2(x,y) - \mathbf{I}_1(x-u, y-v)| + |\mathbf{I}_2(x',y') - \mathbf{I}_1(x'-u, y'-v)|]/7 \\ &-|\mathbf{I}_2(x,y) - \mathbf{I}_2(x',y')|/10\}.\end{aligned}$$

The reassignment probability $q(l|R, \mathbf{X}^{(0)})$ is chosen as in eqn. (17).

At the region level $\mathbf{X}^{(1)}$, the edge weight between two adjacent nodes v, v' (each being a set of pixels) is based on the KL divergence between their intensity histograms h_u, h_v, as in Section 4. The reassignment probability $q(l|R, \mathbf{X}^{(1)})$ is also defined in terms of intensity histograms, following (34).

$$q^{(2)}(v, v') = \exp\{-\frac{1}{2}(KL(h_m(v)||h_m(v')) + KL(h_m(v')||h_m(v))\}. \tag{48}$$

The reassignment probability $q(l|R, \mathbf{X}^{(2)})$ is defined in terms of the motion histograms for each object, similar to (34).



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

Figure 12: 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 gray image.

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

We run the multi-grid and multi-level SW-cut on a number of synthetic and real world motion images. The multi-grid method was only performed on the pixel level, with the window probability inversely proportional to its size. We show four results in Fig. 12. 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. 8 is a fifth

example.

We choose the segmentation example – the cheetah image in Fig. 5 for comparison of the different cluster algorithm methods. In section (4), the pixels are grouped deterministically into atomic regions in a preprocessing stage. Now we perform the cluster algorithm in two levels and the atomic regions are generated by one level of the cluster algorithm process.

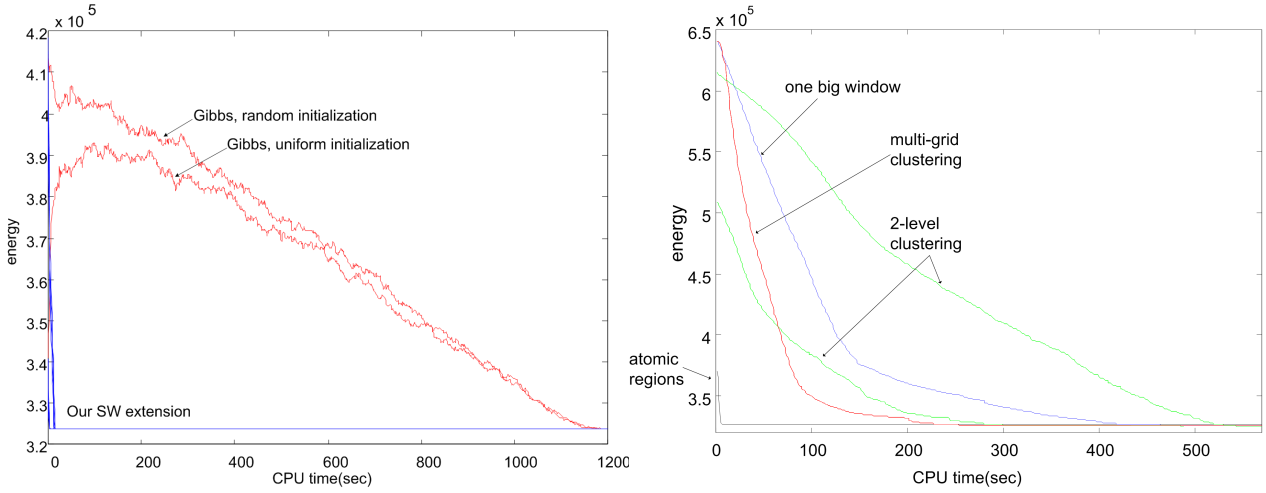


Figure 13: Convergence comparison of multigrid and multi-level cluster algorithm (right) and the Gibbs sampler (left) for the cheetah image in Fig. 5. (see text for explanation)

For the cheetah image, we plot in Fig. 13 the $-\ln \pi(\mathbf{X})$ vs the CPU time for the various methods (right), and of the Gibbs sampler and our algorithm on atomic regions (left). The multi-level cluster algorithm was run in two initializations.

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

Another point to mention is that the multi-grid method is the fastest among the methods that work directly on pixels.

The Gibbs sampler plotted in Fig. 13(left) was run on the deterministic atomic regions not the pixels. If it is run on the pixels, we cannot get it to converge to the minimum in any reasonable time.

7 Discussion

In this paper, we only report the empirical speed of the cluster algorithm 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 to quantify the external fields. In our case, it is impractical to quantify the natural images with a reasonable model. These problems remain open for further investigation.

It is known (Boykov, Veksler and Zabih 2001) that even finding the minimum energy of a Potts model is NP-hard. This suggests that one should not expect to find polynomial time algorithms for solving image analysis applications, but one should look for good suboptimal solutions instead. In this view, one should expect the algorithm presented in this paper to be a good suboptimal solution for graphs with a large number of nodes.

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] Barbu, A. and Zhu S.C. (2005) Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities, *IEEE Trans. on PAMI*, **27**, no. 8, 1239-1253

[4] Barbu, A. (2005). "Cluster sampling and its applications to segmentation, stereo and motion", PhD thesis, UCLA, 2005. http://www.stat.ucla.edu/∼abarbu/thesis.pdf

[5] Besag, J. and Green, P. J. (1993). "Spatial statistics and bayesian computation", *Journal of the Royal Statistical Society, Series B*, **55** no. 1, 25-37

[6] Y. Boykov, O. Veksler, and R. Zabih (2001). "Fast approximate energy minimization via graph cuts". *IEEE Trans. on PAMI*, **23**, no. 11, 1222-1239

[7] 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.

[8] 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.

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

[10] Geyer, C.J. (1991). "Markov chain Monte Carlo maximum likelihood", *Computing Science and Statistics:Proceeding of the 23rd Symposium on the Interface* 156-163.

[11] Geyer, C.J. and Thompson, E.A. (1995). "Annealing Markov chain Monte Carlo with applications to ancestral inference", *J. Am. Statist. Assoc.* **90** 909-920.

[12] 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 .

[13] Goodman, J and Sokal, A.D. (1989) Multigrid Monte Carlo method. Conceptual foundations. *Physical Review D*

[14] 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.

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

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

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

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

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

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

[21] Liu, J.S. (2001). "Monte Carlo strategies in scientific computing", Springer, NY.

[22] 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.

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

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

[25] 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.

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

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