Generalizing Swendsen-Wang to Sampling Arbitrary Posterior Probabilities

Adrian Barbu and Song-Chun Zhu

Abstract

Many vision tasks can be formulated as graph partition problems that minimize energy functions. For such problems, the Gibbs sampler[9] provides general solution but is very slow, while other methods, such as Ncut[24] and graph cuts[4], [22], are computationally effective but only work for specific energy forms[17] and are not generally applicable. In this paper, we present a new inference algorithm that generalizes the Swendsen-Wang method[25] to arbitrary probabilities defined on graph partitions. We begin by computing graph edge weights, based on local image features. Then the algorithm iterates two steps. (i) *Graph clustering*: it forms connected components by cutting the edges probabilistically based on their weights. (ii) *Graph relabeling*: it selects one connected component and flips probabilistically, the coloring of all vertices in the component simultaneously. Thus it realizes the split, merge, and re-grouping of a "chunk" of the graph, in contrast to Gibbs sampler that flips a single vertex. We prove that this algorithm simulates ergodic and reversible Markov chain jumps in the space of graph partitions and is applicable to arbitrary posterior probabilities or energy functions defined on graphs. We demonstrate the algorithm on two typical problems in computer vision - image segmentation and stereo vision. Experimentally we show that it is 100-400 times faster in CPU time than the classical Gibbs sampler and 20-40 times faster then the DDMCMC segmentation algorithm[27]. For stereo, we compare performance with graph cuts and belief propagation. We also show that our algorithm can automatically infer generative models and obtain satisfactory results (better than the graphic cuts or belief propagation) in the same amount of time.

This paper was first published in the Proc. of the 9th Int'l Conf. on Computer Vision, Nice, France, 2003. This version is extended and accepted by IEEE Trans. on PAMI, 2005.

Departments of Computer Science and Statistics, University of California, Los Angeles, Los Angeles, CA 90095. e-mail: abarbu@ucla.edu, sczhu@stat.ucla.edu

I. Introduction

Many computer vision tasks have a "what goes with what" component which can be formulated as a graph partition (or coloring) problem. For example, segmentation and grouping in perceptual organization, and correspondence in stereo and motion. The common objective of these tasks is to partition various image elements, as vertices in an adjacency graph, into a number of coherent visual structures so that a Bayesian posterior probability or an energy function is optimized.

Under the formulation of graph partition, an increasing number of algorithms from computer science and modern statistical physics have been brought to computer vision and become very influential recently. The first prominent method is the graph spectral analysis[32], such as the normalized cuts[24] and its variants for segmentation and grouping that minimize discriminative energy functions. The second popular method is the minimum-cut[22] and the graph cut[4] which map energy minimization problems to maximum flow problems and solve them in low order polynomial time. The third method is the generalized belief propagation on graphs[33], which is shown to minimize some approximate energy functions. All three methods are computationally efficient, but they are limited to specific forms of energy functions and thus not generally applicable in visual inference. We shall address their limitations in comparison to our method later in this section.

For graph partition problems, classic Markov chain Monte Carlo methods, such as Gibbs sampler[9] or "heat bath" in physics, provide general solutions but experience very slow convergence, especially when adjacent vertices in the graph are strongly coupled, i.e. the coloring of the vertices are interlocked locally. Figure 2 illustrates such an example where the Gibbs sampler, which flips the color of a single vertex at each step, has to wait exponentially before changing the color of a set of coupled vertices. The speed problem of Gibbs sampler was addressed by the well-celebrated Swendsen-Wang (SW) method[25], [30]. At each step, the SW algorithm clusters the coupled vertices into connected components each having the same color, and then flips the color of each connected component jointly. For classic Ising/Potts models[19], a new bounding chain technique[14] has been developed recently, and can diagnose the convergence of SW to its invariant probability, i.e. exact sampling, and furthermore the convergence speed (Markov chain mixing time) is polynomial on the graph size n. But the SW method is only valid for Ising/Potts models since the cancelation required in deriving the SW method is not observed in general probabilities or energies. Even worse, SW slows down in the presence of an "external field" (i.e. data or likelihood). More specifically, if one integrates the Potts model as prior probability with likelihood in Bayesian inference, it could be very slow as the graph clustering step does not make use of the data. We shall discuss the SW method and its properties in details in Section (III-B).

In this paper, we generalize SW to arbitrary posterior probabilities or energy functions and derive a generic solution for graph partition. The basic ideas are summarized below.

(i) *Initialization.* Given an adjacency graph, we compute local discriminative probabilities for each edge based on the external field and the prior. For computer vision, local image features or statistical tests are used to obtain these edge weights. Then the algorithm iterates the following two steps.

(ii) *Graph clustering.* Given a current partition (coloring), it removes all edges between vertices of different colors. Then each of the remaining edges, which connects adjacent vertices of the same color, is turned on/off according to its weight. If the discriminative probabilities are informative, then the edges at object boundaries have a high chance to be turned off. Thus it obtains a number of connected components (subgraphs) each having the same color, and usually these connected components correspond to strongly coupled vertices that stand for parts of objects in the image (see Figure 4). We define a "*Swendsen-Wang cut*" for each connected component as the set of edges which connect this component with its neighboring vertices of the same color. In other word, the edges in a Swendsen-Wang cut are turned off probabilistically. These connected components can be regarded as samples from an approximation of the posterior with a Potts model, and they will be accepted by the posterior probability in the next step.

(iii) *Graph flipping.* It selects one (or multiple) connected component and flips, with a probability driven by the posterior, the coloring of all vertices in the selected component(s) simultaneously. Thus it realizes the split, merge, and re-grouping of a "chunk" of the graph, in contrast to the Gibbs sampler that flips a single vertex. The flipping procedure can automatically change the number of colors and thus is more general than the original SW method that works for a fixed number of colors in the Potts model.

We shall show that the new algorithm simulates ergodic and reversible Markov chain jumps in the finite space of all possible graph partitions. The algorithm is valid for sampling arbitrary posterior probability or energy functions.

Our new algorithm mainly makes three contributions. Firstly, we generalize the SW method from the perspective of Metropolis-Hastings method, and derive a simple and analytic formula for the acceptance probability in a reversible Metropolis-Hastings step. This formula (see Theorem 2) is expressed in terms of the product of the discriminative probabilities on the edges (often a very small number) in the Swendsen-Wang cuts. Secondly, we compute the discriminative probabilities on edges from the input image ("external field" in a physics term). We observe that empirically these discriminative probabilities make the connected components more effective, in comparison to a uniform probability in the original SW method. This is in a similar spirit to data-driven Markov chain Monte Carlo[27]. Thirdly, we present various versions of the algorithm. One of the variants is a direct generalization of the Gibbs sampler. It flips the coloring of a connected component according to a conditional probability with a rectifying factor, and the flip is accepted with probability one.

We demonstrate the algorithm on two typical problems in computer vision - image segmentation and stereo vision. In image segmentation, we choose a generative image representation with three classes of image models. It works 100-400 times faster in CPU time than the classic Gibbs sampler and obtains good results in 3-30 seconds in a PC. In the stereo matching problem, we adopt the energy function used in graph cut[4] and the benchmark in[23] for comparison. It obtains good results (better than belief propagation[26]) in 6-10 minutes on a 400x290 image, and is slower than graph cuts. The computing speed certainly depends on the discriminative probabilities in the problem domain. For optimization problems, our method still uses simulated annealing, but at a much quicker schedule than the Gibbs sampler (15 sweeps as opposed to 5000 sweeps) and we do not have to start with a high initial temperature. The algorithm can therefore start with good initial solutions to speed-up convergence.

We now compare our method with other graph partition algorithms in computer vision.

First, it is distinct from the graph spectral analysis[32], such as normalized cuts[24], [32]. We argue that the discriminative energies, used in Ncuts and many other discriminative grouping and clustering algorithms[15], [13], [21], [8], have difficulties in expressing global visual patterns, such as shading effects, perspective projection effects, contour closure etc. Furthermore natural images contain very diverse visual structures which are "coherent" in many different ways, there is no single discriminative criterion that is generally applicable to correctly partition all the visual structures in images[8]. For example, a criterion that prefers compact regions will break elongated curve patterns. Thus we need a generative and Bayesian formulation incorporating a number of diverse and competing image models. Each family of models explains how a pattern is generated and stands for a coherence criterion. For example, seven families of models are used for texture, color, shading and clutter regions in image segmentation[27]. Our algorithm uses the Ncut type discriminative probabilities on edges, but only for making proposals, which are accepted or rejected by the Bayesian posterior probability that incorporates many families of image models and global prior knowledge.

Second, although the graph cut and minimum-cut algorithms[22], [17] are effective in minimizing some energy functions, it is shown [17] that only very limited classes of energy functions can be mapped into the maximum flow problems. For example, so far these methods have not been applicable to generative models with multiple classes of image models.

Third, our method is an addition to the recent data-driven Markov chain Monte Carlo (DDMCMC) algorithm for segmentation[27] and parsing[28] which solves Bayesian inference by mixing a number of reversible jumps. The jumps are divided into two types. Type I solves the "what is what" sub-tasks, such as model selection, switching, and fitting. The DDMCMC algorithm computes discriminative models, such as color and texture clustering, and expresses them in the form of non-parametric probabilities to drive these jumps. Type II solves the "what goes with what" sub-tasks such as grouping, segmentation, and correspondence. Our Swendsen-Wang cut algorithm in this paper improves the type II jumps in both theoretical formulation and computational speed. It can speeds up the DDMCMC algorithm[27] by 20-40 times for segmentation.

In this paper we shall focus on the type II reversible jumps in the graph partition space. We omit discussion on the model spaces for type I jumps, which are referred to [27].

The paper is organized in the following. We present the Bayesian formulation for graph partition in Section (II). Then we discuss the difficulties in sampling the graph partitions and introduce the original SW algorithm in Section (III). Section (IV) presents the new Swendsen-Wang cut algorithm and its variants. Then we show two groups of experiments in Section (V) – image segmentation and stereo matching. Finally Section (VI) concludes the paper with discussions on the advanced topics on extending and analyzing the Swendsen-Wang cuts.

II. Bayesian formulation of graph partition

A. Bayesian formulation

We consider an adjacency graph $G_o =< V, E_o >$, where $V = \{v_1, v_2, ..., v_N\}$ is the set of nodes that need to be partitioned, such as atoms, pixels, edge elements, image primitives, or atomic regions with nearly constant intensities, and E_o is a set of edges connecting neighboring nodes. An n-partition of the graph is denoted by

$$\pi_n = (V_1, V_2, ...V_n), \quad \cup_{i=1}^{n} V_i = V, \ V_i \cap V_j = \emptyset, \ \forall i \neq j. \tag{1}$$

Since visual structures are coherent in many different ways, each subset $V_i, i = 1, 2, ..., n$ is assigned a color $\mathbf{c}_i$ which represents the model, usually consisting of a type (constant, spline, etc) and some parameters. Our objective is to compute the following world representation W from the input $\mathbf{I}$,

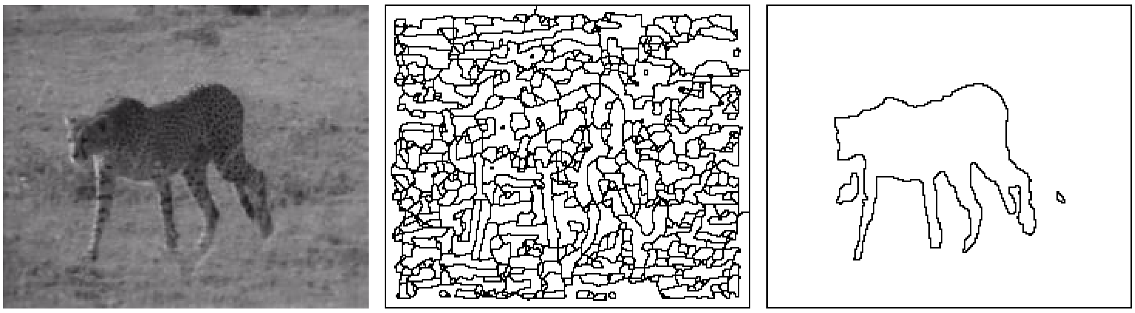
$$W = (n, \pi_n, \mathbf{c}_1, ..., \mathbf{c}_n) \tag{2}$$

This becomes an optimization problem either maximizing the Bayesian posterior probability or minimizing an energy in a solution space Ω,

$$W^* = \arg\max_{W \in \Omega} p(\mathbf{I}|W)p(W), \quad \text{or} \quad W^* = \arg\min_{W \in \Omega} \mathcal{E}(W|\mathbf{I}). \tag{3}$$

We choose two typical vision problems as examples in this paper. We denote by $\mathbf{I}_v$ the image attributes on vertex v, and $\mathbf{I} = \mathbf{I}_V$ the attributes for the set V.

The first example is image segmentation, as shown in Figure 1. Each vertex v is an atomic region with nearly constant intensity, and $\mathbf{I}_v$ is its intensity. A partitioned subset V_i corresponds to a coherent region R_i with model $\mathbf{c}_i = (\ell_i, \theta_i)$ where ℓ_i is the type of image model, and θ_i the model parameters. We adopt three types of simple image models and a prior probability in Section (V-A). Usually, these models should be color, texture, and



a. Input image b. atomic regions in G_o c. segmentation

Fig. 1. Image segmentation as graph partition. a. Input image. b. Atomic regions by Canny edge detection followed by edge tracing and contour closing, each being a vertex in the graph G_o. c. Segmentation result.

shading, as implemented in DDMCMC[27]. Thus the likelihood for $\mathbf{I}_{V_i}$ is $p(\mathbf{I}_{V_i}; \ell_i, \theta_i)$, where θ_i may have different dimensions for different types of models.

The second example is stereo matching. The graph G_o is the pixel lattice, $\mathbf{I}_v = (\mathbf{I}^l_v, \mathbf{I}^r_v)$ is the left and right image intensity and $\mathbf{c}_i$ is the disparity of V_i, discretized along the epipolar line as $\mathbf{c}_i \in \{0, ..., d_{\max}\}$. The energy function is formulated in Section (V-C).

In our recent work[2], we have applied the same SW-cuts algorithm to motion where $\mathbf{c}_i = (u_i, v_i)$ is the motion velocity, or even $\mathbf{c}_i$ can be a vector that includes both motion and image segmentation. Our algorithm has also been used for curve grouping.

B. Solution space and Markov chain jumps

In this subsection, we consider the structures of the solution space and the necessary Markov chain design for optimization in this space. Then we present the place of graph partition in this optimization.

For W in eqn. (2), we denote by $\Omega_{\pi_n} \ni \pi_n$ the space of all possible n-partitions π_n of V, $\Omega_\ell \ni \ell_i$ the set of types of image models, and $\Omega_{\theta_i} \ni \theta_i$ the model parameter space (family) for type ℓ_i. Thus the solution space for W is

$$\Omega = \cup_{n=1}^{N} \{\Omega_{\pi_n} \times \Omega_\ell^n \times \Omega_{\theta_1} \times \cdots \times \Omega_{\theta_n}\}. \tag{4}$$

The factorization of the space corresponds to the two types of moves necessary for exploring the entire space.[1]

1. Type I is "what is what" moves for selecting the model $\ell_i \in \Omega_\ell$ and fitting the model parameters $\theta_i \in \Omega_{\theta_i}$ for $V_i, i = 1, 2, ..., n$. Model fitting is omitted in the stereo matching experiment. We usually can quantize the model spaces so that they become finite.

2. Type II is "what goes with what" moves for grouping, segmentation and correspondence in the partition space $\Omega_\pi = \cup_{n=1}^{|V|} \Omega_{\pi_n}$, which is a finite space.

The two types of moves are tightly coupled, and we implement them by a number of reversible jumps which simulate Markov chain searches in the space Ω. The Markov chain starts with an initial solution W_o and is designed to have a unique invariant (stationary) probability $p(W|\mathbf{I})$. Suppose we denote the state probability of the Markov chain at time t by $p_t(W_o, W)$. A classic measure of convergence is the total variation,

$$||p_t(W_o, W) - p(W|\mathbf{I})||_{\mathrm{TV}} = \frac{1}{2} \sum_{W \in \Omega} |p_t(W_o, W) - p(W|\mathbf{I})|. \quad (5)$$

A measure of the speed of an algorithm $\mathcal{A}$ is the mixing rate, that is the minimum time for the Markov chain to come close to the stationary probability for any W_o,

$$\tau_{\mathcal{A}} = \max_{W_o} \min\{t : \ ||p_t(W_o, W) - p(W|\mathbf{I})||_{\mathrm{TV}} \leq \epsilon\}. \quad (6)$$

Usually $\tau_{\mathcal{A}} = \tau_{\mathcal{A}}(\epsilon, |G_o|)$ is a function of $\frac{1}{\epsilon}$ and the graph size $|G_o|$, i.e. number of vertices and edges. The algorithm $\mathcal{A}$ is said to be rapid mixing if $\tau_{\mathcal{A}}$ is polynomial or logarithmic.

In this paper we shall only study the type II moves and omit the type I moves which have been discussed in the DDMCMC algorithm[27].

[1] It is interesting to note that human brain mapping study[29] shows that the recognition task (type I) is handled by a dorsal stream and the spatial vision (type II) is processed by a ventral stream.

III. Gibbs sampler, Swendsen-Wang method, and their limitations

In this section we discuss the Gibbs sampler and the original Swendsen-Wang algorithm for graph partition to set the background.

A. The difficulty of graph partition by Gibbs sampler

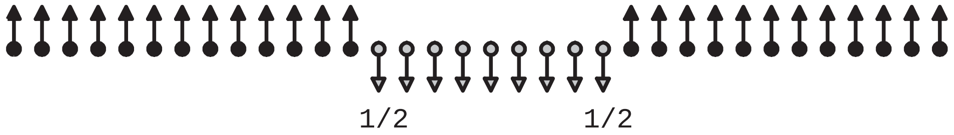


Fig. 2. Difficulty in sampling the Ising and Potts models.

The difficulty of sampling in the partition space Ω_π is well reflected in a simple Ising and Potts model[19], which are sometimes used in vision as prior model. Figure 2 shows a string of spins whose label (color) $\mathbf{c}$ can be $+1$ (up) and -1 (down). A Potts model may have $Q \geq 3$ colors, $\mathbf{c} \in \{1, 2, ..., Q\}$. The Ising/Potts model is

$$p(\pi_n) = \frac{1}{Z} \exp\{\beta \sum_{<s,t> \in E_o} \mathbf{1}(\mathbf{c}_s = \mathbf{c}_t)\}, \qquad \beta > 0. \tag{7}$$

where $\mathbf{1}(\mathbf{c}_s = \mathbf{c}_t) = 1$ if $\mathbf{c}_s = \mathbf{c}_t$ for two adjacent vertices s, t otherwise it is zero. Obviously the highest probability is achieved when all vertices have the same label. In a best visiting scheme, suppose a single site update algorithm, like the Gibbs sampler, flips the -1 spins at the two "cracks" in figure 2. The probability for flipping each spin from -1 to $+1$ is $p_o = 1/2$. Thus to flip a string of k spins ($k = 9$ in Figure 2) from -1 to $+1$ successfully, the expected number of steps is $\frac{1}{(1/p_o)^k} = 2^k$. This is exponential waiting and is typical for general graph partition! Intuitively it will be desirable to flip a big set of vertices that have the same color at each step. Of course, we need to ensure that such moves still keep $p(\pi_n)$ as its stationary probability. This is what the Swendsen-Wang method does.

B. Swendsen-Wang on Potts models and theoretical results

There are many ways to interpret the SW algorithm, including random cluster model, auxiliary variables[6] and slice sampling and decoupling[12]. In this paper, we interpret the SW method as a Metropolis-Hastings step and our interpretation leads to generalizing it to arbitrary probabilities in Section (IV).

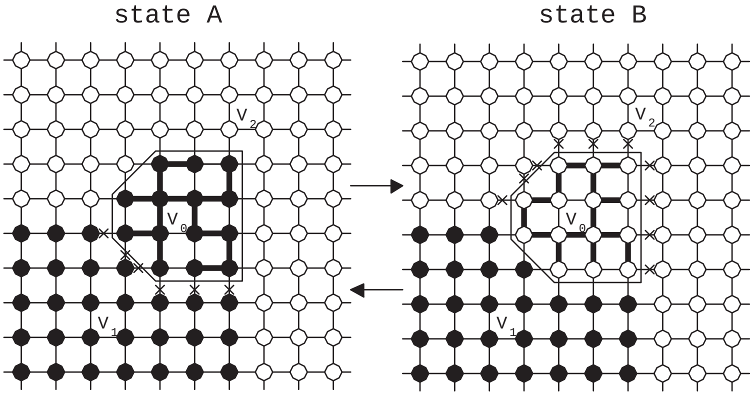


Fig. 3. SW algorithm flips the color of a set of vertices V_o in one step for the Ising/Potts models. The set of edges marked with crosses is called the Swendsen-Wang cut.

Consider a Potts model in eqn. (7) on a 2D lattice. Figure 3 shows two partition states π_A and π_B with $\pi_A = (V_o \cup V_1, V_2, \cdots)$ and $\pi_B = (V_1, V_o \cup V_2, \cdots)$, which differ by the labels of the vertices V_o inside the center window.

The SW algorithm realizes a reversible move between π_A and π_B in a single step. From state π_A, the SW algorithm proceeds in the following way:

1. Any edge $e =< s, t >\in E_o$, is removed if $\mathbf{c}_s \neq \mathbf{c}_t$. If $\mathbf{c}_s = \mathbf{c}_t$, then $e =< s, t >$ is turned "on" with a probability $q_o = 1 - e^{-\beta}$ otherwise it is turned "off", i.e. removed. This yields a number of connected components, each being a subset of vertices of the same color.

2. It randomly selects a connected component V_o of the resulting graph (see Figure 3 (left)). The dark edges in V_0 remain **on**, the other edges have been turned **off**.

3. It chooses a label $\mathbf{c} \in \{1, , ..., Q\}$ for V_o with uniform probability.

In the example of Figure 3, V_o change color from black to white and we obtain partition state π_B in Figure 3 (right). Reversely, at state π_B, we will have a chance to select V_o and flip it to black color and this return to π_A.

In this paper, the *Swendsen-Wang cuts* at π_A and π_B are the sets of edges connecting V_o to V_1 and V_2 respectively, marked by the crosses in Figure 3.

$$\mathcal{C}_A = \mathcal{C}(V_o, V_1) = \{(s,t) : s \in V_o, t \in V_1\}, \quad \mathcal{C}_B = \mathcal{C}(V_o, V_2) = \{(s,t) : s \in V_o, t \in V_2\}. \tag{8}$$

In state π_A, there is a combinatorial number of ways to make V_0 a connected component, but in all cases, the edges in $\mathcal{C}_A$ must have been cut probabilistically. Similarly in state π_B, edges in $\mathcal{C}_B$ must be turned off in order for V_o to be a connected component.

We look at the moves between states π_A and π_B from the perspective of the Metropolis-Hastings method[18]. Though it is computationally difficult to compute the proposal probabilities $q(\pi_A \to \pi_B)$ and $q(\pi_B \to \pi_A)$, one can compute their ratio easily through cancelation.

$$\frac{q(\pi_A \to \pi_B)}{q(\pi_B \to \pi_A)} = \frac{(1-q_o)^{|\mathcal{C}_A|}}{(1-q_o)^{|\mathcal{C}_B|}} = (1-q_o)^{|\mathcal{C}_A|-|\mathcal{C}_B|}. \tag{9}$$

$\mathcal{C}_A$ is the cardinality of set $\mathcal{C}_A$. Remarkably the probability ratio for $p(\pi_A)/p(\pi_B)$ for the Potts model is also decided by the Swendsen-Wang cuts

$$\frac{p(\pi_A)}{p(\pi_B)} = \frac{e^{-\beta|\mathcal{C}_B|}}{e^{-\beta|\mathcal{C}_A|}} = e^{\beta(|C_A|-|C_B|)} \tag{10}$$

The acceptance probability for the move from π_A to π_B is,

$$\alpha(\pi_A \to \pi_B) = \min(1, \frac{q(\pi_B \to \pi_A)}{q(\pi_A \to \pi_B)} \cdot \frac{p(\pi_B)}{p(\pi_A)}) = (\frac{e^{-\beta}}{1-q_o})^{|\mathcal{C}_A|-|\mathcal{C}_B|} = 1 \tag{11}$$

if we take $q_o = 1 - e^{-\beta}$, so the proposal from π_A to π_B is always accepted. So once V_o is selected, its new color is picked at random without having to go through the Metropolis-Hastings step due to the cancelation! As $\beta \propto \frac{1}{T}$ is the inverse of the "temperature" in the Potts models, at lower temperature, $q_o \to 1$ and SW flips a larger patch each time.

For Potts models in eqn. (7), Huber [14] developed a new bounding chain technique[14] which can diagnose the convergence of SW, i.e. exact sampling or perfect sampling[20]. The number of steps in reaching exact sampling is in the order of $O(\log |E_o|)$ for temperature far below and far above the critical temperature. Using a path coupling technique, Cooper and Frieze [5] have showed that the mixing time τ (see eqn. (6)) is polynomial[5] if each vertex in graph G_o is connected to $O(1)$ number of neighbors, i.e. the connectivity of each vertex does not grow with the size of V. This is usually observed in vision problems, such as the lattice. The mixing time becomes exponential at a worst case when G_o is fully connected[10]. Such case may never occur in vision problems.

However the excitement of SW algorithm has been limited for the following reasons.

1. It is restricted to Ising and Potts models, while posterior probabilities in vision tasks are of much more complex forms.
2. It becomes very slow even for the Potts models in the presence of external fields (data). As q_o is a constant, it does not utilize the input data in clustering the connected components.
3. It assumes the number of labels n is fixed. The Markov chain does not create new labels in cases where n is unknown (in vision, usually the number of models is unknown).

In the next section, we overcome these limitations and extend SW to arbitrary probabilities.

IV. Graph partition by Swendsen-Wang cuts

A. Discriminative probabilities on edges

Before running the reversible jumps, we augment the adjacency graph $G_o = < V, E_o >$ with discriminative probabilities in an initial stage. Partition samples obtained using these probabilities will be used in the next subsection as proposals for the full posterior probability. For any vertex $v \in V$, we extract a number of features $F(v) = (F_1(v), F_2(v), ..., F_a(v))$, and for each edge $e = < s, t > \in E_o$ we assign a binary random variable $\mu_e \in \{\text{on}, \text{off}\}$. μ_e indicates whether the edge is turned on or off. Then we compute a discriminative probability $q_e = q(\mu_e = \text{on}|F(s), F(t))$ based on local features $F(s)$ and $F(t)$.

Take the adjacency graph in Figure 1 as an example. For each atomic region (vertex in G_o), we compute a 15-bin intensity histogram h normalized to 1. For each edge $e = < v_i, v_j >$, we define

$$q_e = p(\mu_e = \text{on}|h_i, h_j) = e^{-(KL(h_i||h_j)+KL(h_j||h_i))T/2}, \tag{12}$$

where $KL()$ is the Kullback-Leibler divergence between the two histograms and T is a temperature factor. Usually q_e should be close to zero for e on object boundary. Suppose we turn on the edges independently according to $q_e, e \in E_o$, we obtain a sparse graph $G = < V, E >$ with probability

$$q(E) = \prod_{e \in E} q_e \prod_{e \in E_o \setminus E} (1 - q_e). \tag{13}$$

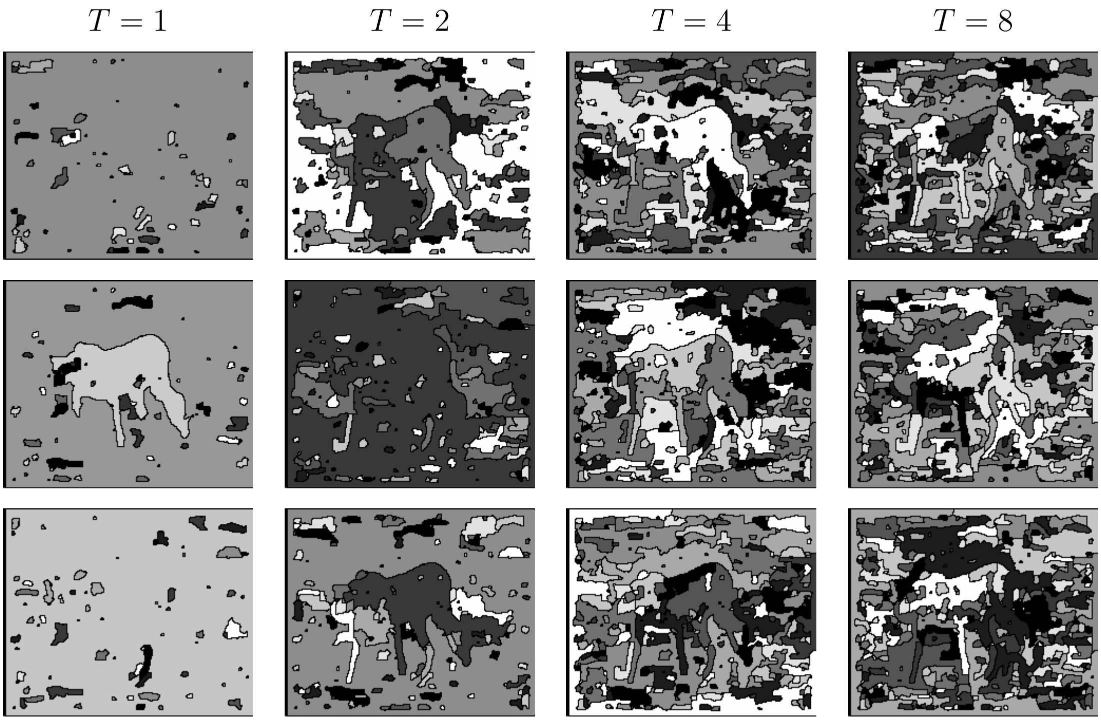
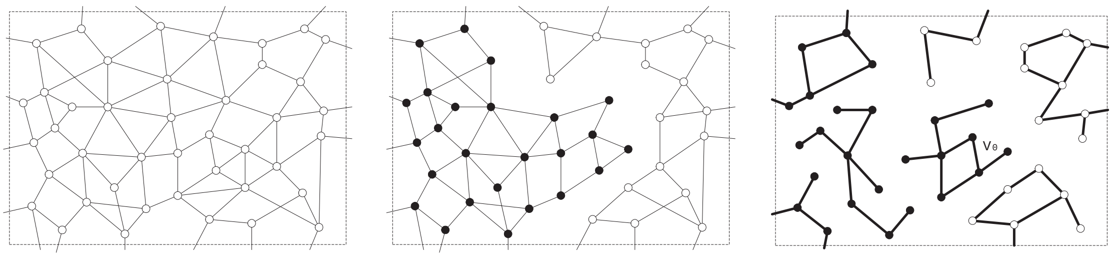


Fig. 4. Random clustering of the adjacency graph using independent discriminative models on edges. Each uniform region is a connected component.

Then G consists of a number connected components. Figure 4 shows some examples of G for G_o in Figure 1. Each region of uniform grey level is a connected component that consists of a number of atomic regions. We show three random partitions sampled according to $q(E)$ for four temperatures $T = 1, 2, 4, 8$. At a reasonable temperature, various parts of the cheetah are obtained, legs, body and tail, as connected components, which are then proposed as candidates for partition in the reversible jumps.

This example shows that the discriminative models are good heuristics for partition. However, these partitions are limited by the local features. More complex posterior probabilities with global generative models are needed to accept these proposals, and this is done next.

B. Swendsen-Wang cuts and its variants



a. G_o b. G c. CP

Fig. 5. Three stages of graphs in the algorithm. a) Adjacency graph G_o, b. graph G for current partition (coloring) π, c. connected components CP by turning off some edges in G.

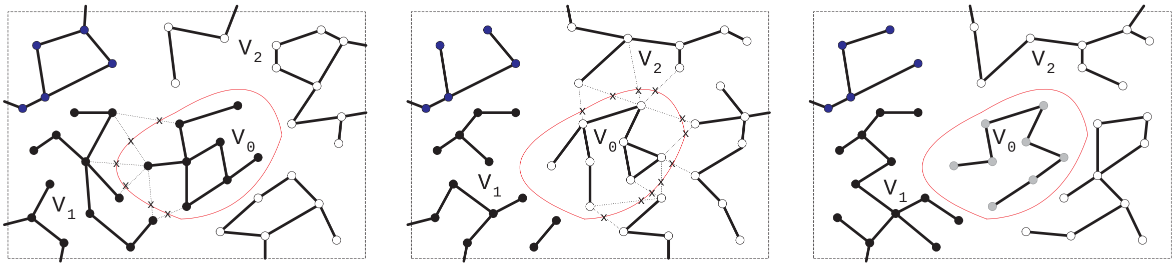
The Swendsen-Wang cut algorithm engages three types of graphs shown in Figure 5. It starts with an adjacency graph $G_o =< V, E_o >$ (Fig.5.a). At each time step we have a partition $\pi = (V_1, ..., V_n)$ which assigns a color to each vertex $\mathbf{c}_v = \ell$ for $v \in V_\ell, \ell = 1, 2, ..., n$, and we obtain a graph $G(\pi) =< V, E(\pi) >$ (Fig.5.b) with $E(\pi) = \{e =< s, t >: \mathbf{c}_s = \mathbf{c}_t\}$. Then each edge $e \in E(\pi)$ is turned off with probability $1 - q_e$ independently, and we obtain a sparse graph CP with a number of connected components.

Now we present a first version of the Swendsen-Wang cut algorithm.

Swendsen-Wang Cut: SWC-1

```
*Input*: $G_o =< V, E_o >$, $q_e, \forall e \in E_o$, and posterior $p(W|\mathbf{I})$.    *Output*: Samples $W \sim p(W|\mathbf{I})$.
1. Initialize a partition $\pi$ by random clustering (*see Fig. 4*)
2. Repeat, for current state $\pi = (V_1, V_2, ..., V_n)$,
3.     For $e \in E(\pi)$, turn $\mu_e =$ off with probability $1 - q_e$.
4.     $V_\ell = (V_{\ell 1}, ..., V_{\ell n_\ell})$ is divided into $n_\ell$ connected components for $\ell = 1, 2, ..., n$.
5.     Collect all the connected components in set $CP = \{V_{\ell i} : \ell = 1, ..., n, i = 1, ..., n_\ell\}$.
6.     Select a connected component $V_o \in CP$ with prob. $q(V_o \,|\, CP)$, say $V_o \subset V_\ell$.
           (*Usually* $q(V_o \,|\, CP) = \frac{1}{|CP|}$ *is uniform, Fig.6.a is an example of* $V_o$ *in partition* $\pi = \pi_A$).
7.     Propose to assign $V_o$ a new label $\mathbf{c}_{V_o} = \ell'$ with a probability $q(\ell'|V_o, \pi)$, thus obtain $\pi'$
           ($\pi' = \pi_B$ *is in Fig.6.b if* $V_0$ *is merged to an existing color* $V_2$,
           *or* $\pi' = \pi_C$ *is in Fig.6.c if* $V_o$ *is assigned a new color*).
8.     Accept the proposal with probability $\alpha(\pi \rightarrow \pi')$ defined in theorem 2.
```

The proposal probability $q(l'|V_o, \pi)$ can be uniform, or better, dependent on the similarity of V_o with $V_{l'}$. At each step, model switching and fitting (type I jumps) are performed deterministically or sampled from some proposal probabilities (see in later this section).



a. A CP of state π_A b. A CP of state π_B c. A CP of state π_C

Fig. 6. A reversible move between three partition states π_A, π_B, π_C which differ only in the color of V_0. The vertices connected by thick edges form a connected component. The thin lines marked with crosses are edges in the SW-cuts.

In the above algorithm, let $V_o \subseteq V_\ell$ in π, and $V_o \subseteq V_{\ell'}$ in π'. The move $\pi \longrightarrow \pi'$ can realize three types of moves depending on the choice of the new color of V_o. Thus the number of color n will be decided automatically.

1. *re-grouping*: $V_o \subset V_\ell$ is split from V_ℓ and merged into an existing color $V_{\ell'}$. The number

of colors n is unchanged. E.g. $\pi_A \leftrightarrow \pi_B$ in Fig. 6. When V_ℓ and $V_{\ell'}$ are adjacent, this is in fact a discrete version of the boundary evolution, like region competition[34].

2. *Splitting*: $V_o \subset V_\ell$ is split into a new color $\ell' = n+1$. E.g. $\pi_A \to \pi_C$ in Fig. 6.

3. *Merging*: $V_o = V_\ell$ and is merged into an existing color. E.g. $\pi_C \to \pi_A$ in Fig. 6.

The second version of the algorithm differs only in the way it selects the set V_o. Instead of sampling all the edges in a current partition, it starts from a single vertex (seed) v and grows into a connected component V_o.

Swendsen-Wang Cuts: SWC-2

```
1. Repeat, for current state $\pi = (V_1, V_2, ..., V_n)$,
2.     Select a seed vertex $v$, say $v \in V_\ell$ in $\pi$. Set $V_o \leftarrow \{v\}$, $\mathcal{C} \leftarrow \emptyset$,
3.     Repeat until $\mathcal{C} \cap \mathcal{C}(V_o, V_\ell \setminus V_o) = \mathcal{C}(V_o, V_\ell \setminus V_o)$,
4.         For any $e =< s, t >\in \mathcal{C}(V_o, V_\ell \setminus V_o)$, $s \in V_o,\ t \in V_\ell \setminus V_o$.
5.             Turn $\mu_e =$ on with probability $q_e$, else $\mu_e =$ off,
6.             If $\mu_e =$ on, set $V_o \leftarrow V_o \cup \{t\}$, else $\mathcal{C} \leftarrow \mathcal{C} \cup \{e\}$.
7.     Propose to assign $V_o$ a new label $\ell'$ with prob. $q(\mathbf{c}_{V_o} = \ell' | V_o, \pi)$.
8.     Accept the move with probability $\alpha(\pi \to \pi')$ defined in theorem 2.
```

Now we compute the acceptance probability $\alpha(\pi \to \pi')$ in SWC-1 and SWC-2.

We start with computing the probability ratio for selecting V_o in $\pi \to \pi'$ and $\pi' \to \pi$.

Theorem 1: Let π and π' be a pair of reversible partition states which differ in the coloring of V_o, with $V_o \subseteq V_\ell$ in π and $V_o \subseteq V_{\ell'}$ in π', then

$$\frac{q(V_o|\pi)}{q(V_o|\pi')} = \frac{\prod_{e \in \mathcal{C}(V_o, V_\ell \setminus V_o)}(1-q_e)}{\prod_{e \in \mathcal{C}(V_o, V_{\ell'} \setminus V_o)}(1-q_e)}. \tag{14}$$

$\prod_{e \in \mathcal{C}(V_o, V_\ell \setminus V_o)}(1-q_e) = 1$ if $V_\ell \setminus V_o = \emptyset$.

Proof: See Appendix A. This is the most important step in obtaining the acceptance probability. It states the fact that although there are a combinatorial number of ways for selecting V_o in π and π', their probability ratio is simple due to cancellations. ■

Theorem 2: In the above notation, the acceptance probability for move $\pi \longrightarrow \pi'$ is

$$\alpha(\pi \to \pi') = \min(1, \frac{\prod_{e \in \mathcal{C}(V_o, V_{\ell'} \setminus V_o)}(1 - q_e)}{\prod_{e \in \mathcal{C}(V_o, V_\ell - V_o)}(1 - q_e)} \cdot \frac{q(\mathbf{c}_{V_o} = \ell | V_o, \pi')}{q(\mathbf{c}_{V_o} = \ell' | V_o, \pi)} \cdot \frac{p(\pi'|\mathbf{I})}{p(\pi|\mathbf{I})}). \tag{15}$$

Proof: By Metropolis-Hastings method[18], the acceptance probability is,

$$\alpha(\pi \to \pi') = \min(1, \frac{q(\pi' \to \pi)}{q(\pi \to \pi')} \cdot \frac{p(\pi'|\mathbf{I})}{p(\pi|\mathbf{I})}), \tag{16}$$

For the re-grouping case (see $\pi_A \leftrightarrow \pi_B$ in Fig. 6), there is only one path moving between the two states π and π', i.e. selecting and flipping V_o. Therefore,

$$\frac{q(\pi' \to \pi)}{q(\pi \to \pi')} = \frac{q(V_o|\pi')}{q(V_o|\pi)} \cdot \frac{q(\mathbf{c}_{V_o} = \ell | V_o, \pi')}{q(\mathbf{c}_{V_o} = \ell' | V_o, \pi)}. \tag{17}$$

The conclusion follows straight from theorem 1. For the splitting and merging case (see $\pi_A \leftrightarrow \pi_C$ in Fig. 6), there are two paths. We put the proof in Append B for clarity. ■

As the partition space $\Omega_\pi \ni \pi$ is finite, the Markov chain in SWC-1, 2 is then ergodic following the observation that there is a non-zero probability for any node $v \in V$ to be chosen as V_o and assigned a new color. Then the Markov chain can move from a partition to any other partition with non-zero probability in $|V|$ steps.

To include the type I moves for model selecting and fitting, we augment the move from two partitions $\pi \leftrightarrow \pi'$ to two states $W \leftrightarrow W'$. In state W, the set $V_\ell \supseteq V_o$ has image model θ_ℓ the set $V_{\ell'}$ has image model $\theta_{\ell'}$. In state W', V_o is split from V_ℓ and merged into $V_{\ell'}$. The set $V_\ell \setminus V_o$ has a new model θ'_ℓ, and the set $V_{\ell'} \cup V_o$ has model $\theta'_{\ell'}$, obtained by sampling from proposals $q(\theta'_\ell|\mathbf{I}_{V_\ell \setminus V_o}), q(\theta'_{\ell'}|\mathbf{I}_{V_{\ell'} \cup V_o})$ respectively. Then the acceptance probability is

$$\alpha(W \to W') = \min(1, \frac{q(\theta_\ell|\mathbf{I}_{V_\ell})q(\theta_{\ell'}|\mathbf{I}_{V_{\ell'}})}{q(\theta'_\ell|\mathbf{I}_{V_\ell \setminus V_o})q(\theta'_{\ell'}|\mathbf{I}_{V_{\ell'} \cup V_o})} \cdot \frac{q(\pi' \to \pi)}{q(\pi \to \pi')} \cdot \frac{p(W'|\mathbf{I})}{p(W|\mathbf{I})}.$$

The dimensions of the model parameters are matched in the ratio. The proposal probabilities $q(\theta|\mathbf{I}_{V_\ell})$ for any set $V_\ell \in V$ are again a product of discriminative probabilities on the vertices. They are computed in a bottom-up step through data clustering, see [27].

C. SWC-3: generalized Gibbs sampler

Now we design the probability $q(\mathbf{c}_{V_o} = \ell'|V_o, \pi)$ to achieve acceptance probability 1. The third version of our algorithm, named SWC-3, becomes a generalized Gibbs sampler.

Let $\pi = (V_1, V_2, ..., V_n)$ be the current partition, and $V_o \subseteq V_\ell$ be a connected component whose color $\mathbf{c}_{V_o}$ has $n+1$ choices. That is, V_o can be merged with one of the following sets,

$$S_1 = V_1,\ S_2 = V_2, ..., S_l = V_\ell \setminus V_0, ...,\ S_n = V_n, S_{n+1} = \emptyset. \tag{18}$$

By assigning $\mathbf{c}_{V_o} = \ell' \in \{1, 2, ..., n+1\}$, we have $n+1$ possible partitions for π' (n if $V_l \backslash V_o = \emptyset$, and we denote them by $\pi_1, \pi_2, ..., \pi_{n+1}$ respectively. V_o is merged with S_i in π_i for $i = 1, 2, ..., n$ and $\pi_\ell = \pi$. These partitions may have $m = n-1, n, n+1$ colors. We use m for clarity of notation.

We denote the Swendsen-Wang cuts between V_o and $S_j, j = 1, 2, ..., n+1$ by

$$\mathcal{C}_i = \mathcal{C}(V_o, S_i), i = 1, 2, ..., n+1, \quad \text{with } \mathcal{C}(V_o, \emptyset) = \emptyset,\ \cup_{i=1}^{n} \mathcal{C}_i = \mathcal{C}(V_o, V \setminus V_o). \tag{19}$$

The number of edges in these SW-cuts is fixed regardless the number of colors m. We denote the weight for the $n+1$ partitions by

$$\omega_i = \prod_{e \in \mathcal{C}_j} (1 - q_e), \quad i = 1, 2, ..., m. \tag{20}$$

Theorem 3: Given the partition of $V \setminus V_o$, and let $p(\pi_i|\mathbf{I})$ be the posterior probability of partition π_i for $i = 1, 2, ..., m$, if we choose the new color or V_o by

$$q(\mathbf{c}_{V_o} = i|V_p, \pi) = \frac{\omega_i p(\pi_i|\mathbf{I})}{\sum_{j=1}^{m} \omega_j p(\pi_j|\mathbf{I})} \tag{21}$$

then the proposed move is accepted with probability one.

Proof: . For any two partitions π_ℓ and $\pi_{\ell'}$, we have the following acceptance probability, from theorem 2,

$$\alpha(\pi_\ell \to \pi_{\ell'}) = \min(1, \frac{\omega_{\ell'}}{\omega_\ell} \cdot \frac{\omega_\ell p(\pi_\ell|\mathbf{I})}{\omega_{\ell'} p(\pi_{\ell'}|\mathbf{I})} \cdot \frac{p(\pi_{\ell'}|\mathbf{I})}{p(\pi_\ell|\mathbf{I})}) = 1. \tag{22}$$

because the denominator in eq. 21 is the same for π_l and $\pi_{l'}$. ■

Intuitively, once we pick up V_o, we merge V_o with S_i according to the posterior probability $p(\pi_i|\mathbf{I}), i = 1, 2, ..., m$ modified by the SW-cut factor $\omega_i = \prod_{e \in \mathcal{C}_i}(1 - q_e)$ to ensure the Markov chain follows the posterior. If V_o is always a single site, then $\mathcal{C}_i = \emptyset$ and $\omega_i = 1$ for $i = 1, 2, ..., m$, and this reduces to the Gibbs sampler. Now we get the third version of the SWC algorithm

```
Swendsen-Wang Cuts: SWC-3
1. Repeat, for a current partition $\pi = (V_1, ..., V_n)$.
2.         Select a candidate set $V_o$ as in SWC-1 or SWC-2
3.         Draw a random sample $\ell'$ with probability $q(\ell' = i|V_o, \pi)$ from (21)
4.         Merge $V_0$ to $S_i$
```

In comparison, SWC-3 is computationally more costly as it has to evaluate m posteriors at each step. Sometimes we can limit the number of color m to only the sets which are adjacent to V_o. SWC-2 has a smaller computational cost than SWC-1 as it only tests a small number of edges in the graph clustering step. In SWC-2 one can choose the initial seed vertex $v \in V$ according to the goodness of fit, to avoid picking large components every time.

V. Experiments – segmentation and stereo

In this section, we apply the SW-cut algorithms to two classical vision problems– image segmentation and stereo matching.

For optimizing the posterior probability, one needs a simulated annealing procedure[16] that raises the posterior probability to a certain power called temperature. This temperature is slowly decreased according to a cooling schedule. The initial temperature $T_{\max}$ is big, in order to avoid being stuck in local minima, and then it is reduced it to $T_{\min}$ in a given number of sweeps (1 sweep= $|V|$ steps). The initial temperature $T_{\max}$ depends on the efficiency of the algorithm. As Figure 8 shows empirically, the Gibbs sampler needs very high initial temperature and decrease slowly (in 5000 sweeps) in order to reach good solutions. Any good initial solution W_o will be destroyed (randomized) at the high temperature. In comparison the Swendsen-Wang cuts can walk fast at low temperature and we start with $T_{\max}$ small, usually $T_{\max} < 20$, and decrease fast (in 15 sweeps), and it can utilize good initial solutions. The ending temperature $T_{\min}$ is often in the range of $[0.1, 1]$ for our experiments.

A. Experiment I: image segmentation

To reduce the size of the adjacency graph, we use a Canny edge detector and edge tracing to divide the image into "atomic regions" with almost constant intensities. Depending on image size and texture, there are $N \in [500, 1500]$ atomic regions, each being a vertex in G_o. The use of atomic regions helps reduce the computational time, but we should be aware of the risk that we are not able to break them if they are wrong, especially some case of "leakage" occurs and the atomic region becomes rather big. In more recent work[2], we overcome this problem by hierarchic SW-cut method which works on multiple levels of adjacency graphs where the vertices are of various granularities.

We adopt three simple image models and more sophisticated models can be easily added as in [27]. Let x, y be the coordinates of a pixel.

The first model C_1 assumes constant intensity with additive noise modeled by a non-parametric histogram $\mathcal{H}$.

$$\mathbf{J}_1(x, y; \theta) = \mu + \eta, \quad \eta \sim \mathcal{H}, \quad \theta_1 = (\mu, \mathcal{H}). \tag{23}$$

The second model C_2 assumes a linear function with additive noise $\mathcal{H}$. A linear model:

$$\mathbf{J}_2(x, y; \theta) = \mu + ax + by + \eta, \quad \eta \sim \mathcal{H}, \quad \theta_2 = (\mu, a, b, \mathcal{H}). \tag{24}$$

The third model C_3 assumes a quadratic function with additive noise $\mathcal{H}$,

$$\mathbf{J}_3(x, y; \theta) = \mu + ax + by + cx^2 + dxy + ey^2 + \eta, \quad \eta \sim \mathcal{H}, \theta_3 = (\mu, a, b, c, d, e, \mathcal{H}). \tag{25}$$

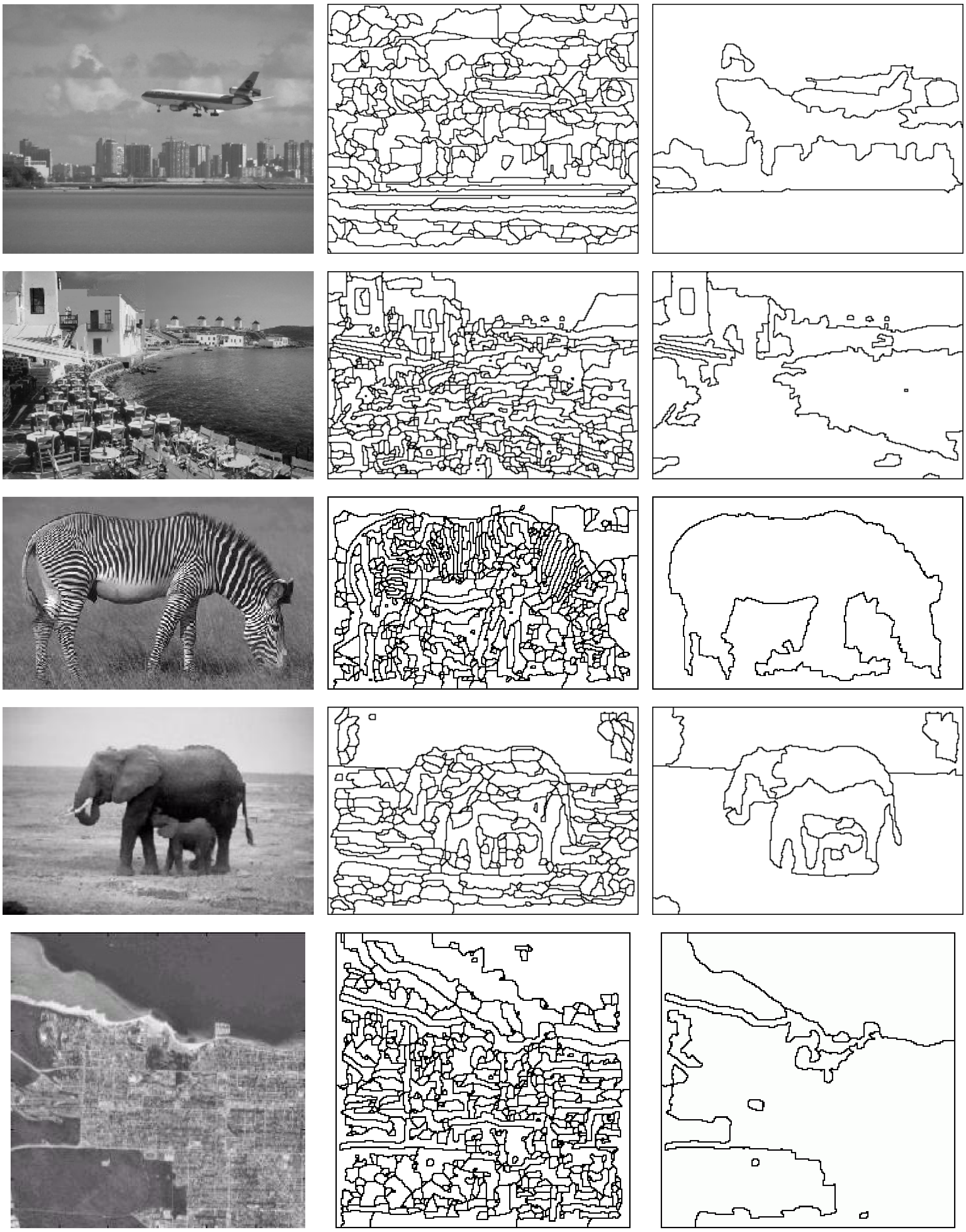


Fig. 7. Image segmentation: input image, atomic regions and segmentation result.

The selection of model was studied in previous DDMCMC work [27]. Such models are found to be useful for fitting smoothness regions with global shading effects. The texture is modeled by the non-parametric histogram $\mathcal{H}$, which in practice is represented by a vector of B-bins $(\mathcal{H}_1, ..., \mathcal{H}_B)$ normalized to sum to 1. Let R be a region which has a model (ℓ, θ). Then the likelihood is

$$P(\mathbf{I}_R;\ \ell, \theta) \propto \prod_{v \in R} \mathcal{H}(\mathbf{I}_v) = \prod_{j=1}^{B} \mathcal{H}_j^{n_j} = \exp(-|R| \text{entropy}(\mathcal{H})). \tag{26}$$

where n_j is the number of pixels of R that fall into the jth bin of the histogram.

Like[27], we use the prior $p(W)$ to encourage large and connected regions. Let n be the number of regions, each region may consist of one or many sub-regions. We denote these connected components by $r_1, r_2, ..., r_m$, $m \geq n$. The prior is

$$p(W) \propto e^{-\gamma n} e^{-\gamma' m} \prod_{i=1}^{m} e^{-\lambda \text{Area}(r_i)^{0.9}}. \tag{27}$$

We fix $\gamma = 35, \gamma' = 15$ in our experiments.

The *model parameters* for the regions are computed deterministically at each step as the best least square fit. This could be replaced by separate steps of model fitting and switching, but this is beyond the purpose of our experiments. The segmentation results obtained from SWC-1 are shown in Figures 1 and 7.

B. Computational speed and comparison

We compare the speed of our algorithm and Gibbs sampler in Figures 8 and 9. We run the SWC-1 algorithm 5 times on the cheetah image in Figure 1, with two types of initializations. One is random initialization which assigns a random color to each atomic region independently with $n = 5$ colors in total. The other is a uniform initialization which sets all atomic regions to the same color $n = 1$. It happens that the uniform initialization has lower energy $(-\log p(W|\mathbf{I}))$ than the random initializations.

To achieve the same low energy level, the Gibbs sampler (upper two curves) in Fig. 8 has to start with a high temperature $T = 200$ and use a logarithmic annealing schedule to $T = 0.05$ in 5000 sweeps, otherwise it remains stuck at a higher energy level. In contrast, the SWC-1 starts at temperature $T = 20$ and decreases to $T = 0.05$ in 15 sweeps. Figure 8 plots the energy for each run as a function of the CPU time in seconds.

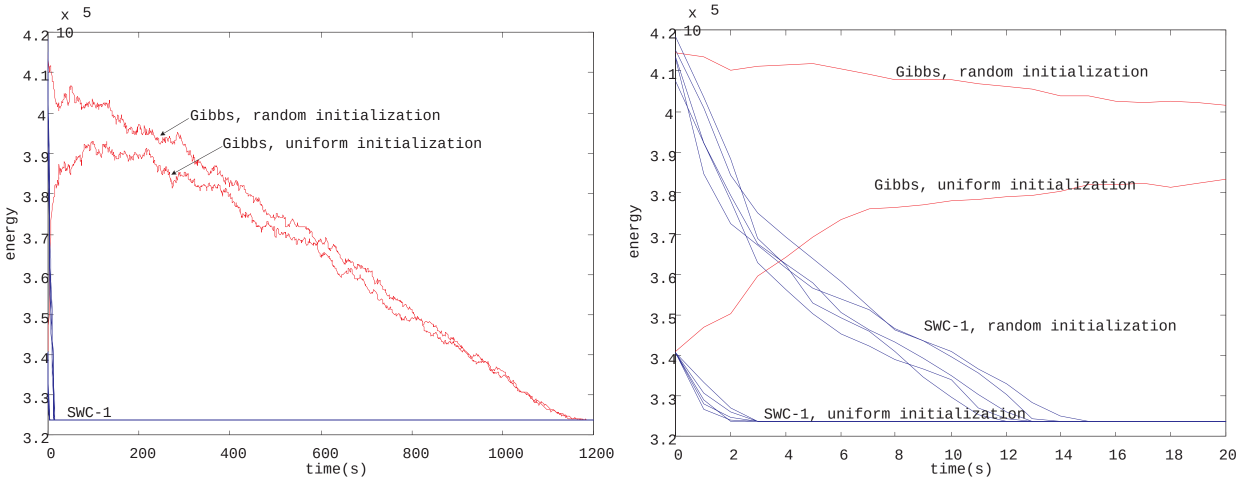


Fig. 8. Convergence comparison between SWC-1 and Gibbs sampler (upper curves) in CPU time (seconds). (Left) The first 1,200 seconds. (Right) Zoomed-in view of the first 20 seconds. The SWC-1 runs 5 times for both the random and uniform initializations.

The two upper curves are the Gibbs sampler with random and uniform initialization respectively. As SWC-1 converges much faster, we plot a zoom-in view of the first 20 seconds. We show 5 SWC-1 runs, for both the random and uniform initializations. The uniform initialization has much lower energy to start with and the SWC-1 algorithm also converges faster (in 3 seconds). In contrast, the Gibbs sampler cannot utilize the good initialization because it has to raise the temperature high.

To study the effects of the discriminative probabilities q_e on convergence speed, we compare the performance of our algorithm with and without discriminative probabilities in Fig.9. We run the SWC-1 algorithm 3 times with all edges having the constant probability, $q_e = 0.2, 0.4, 0.6$ respectively (Note that the Gibbs sampler is equivalent to SWC with $q_e = 0$). The annealing schedules for these runs have to be slower, starting at higher temperature, to obtain the same final energy. Sometimes the algorithm cannot reach the same low energy as with discriminative models.

Fig.9 displays the energy vs CPU time (in seconds) of the three runs and the SWC-1 on the airplane image shown in Fig.7. The energies of the three SWC runs with constant edge probability $q_e = 0.2, 0.4, 0.6$ are shown in dotted lines, all three runs start from a uniform initialization. They are significantly slower than SWC-1. It is worth mentioning that these SWC runs without discriminative probabilities are not equivalent with the original SW algorithm because we work on a more general energy function, on which the original SW cannot be applied.

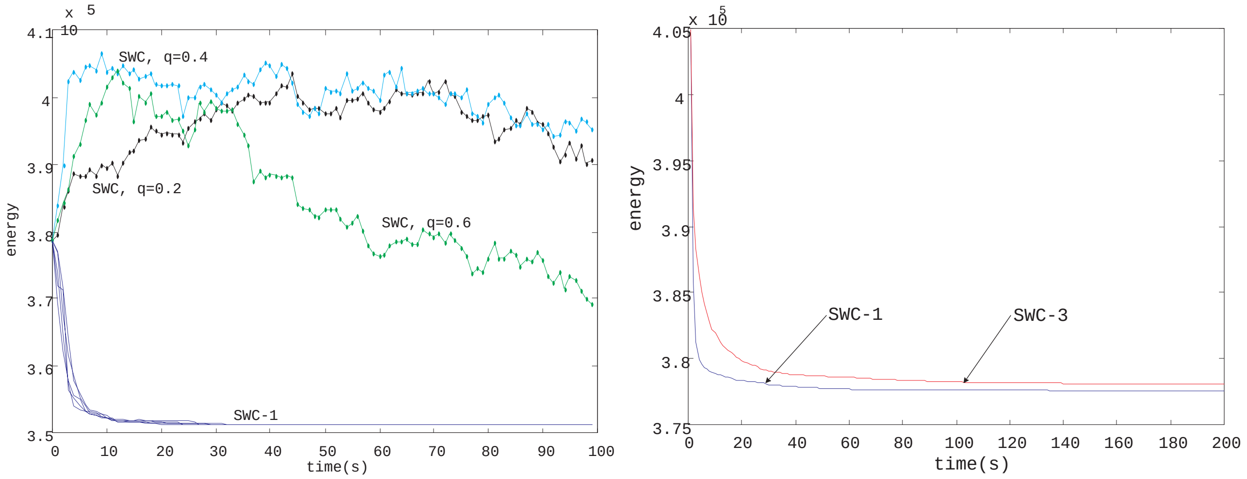


Fig. 9. (Left) Convergence comparison between SWC-1 (solid) and SWC-1 using constant edge probabilities $q_e = 0.2, 0.4, 0.6$ (dotted). (Right) Comparison of SWC-1 and SWC-3 for the second image in Figure 7. Both plots are in CPU time. SWC-3 has more overhead at each step and is slower in this example.

Fig. 9.b compares SWC-1 and SWC-3. SWC-1 is more effective than SWC-3 because of the computational overhead of each SWC-3 move, and that there is more data-driven information used in the SWC-1 than in SWC-3, existent in the design of the $q(l'|V_o, \pi)$.

Compared with the DDMCMC algorithm from [27], our algorithm can speed it up by 20-40 times in CPU time. Our model fitting and switching steps are quite simple, but we observed that the full-featured model fitting and switching steps take much less time than the split-merge steps which are the focus of our algorithm. By incorporating full-featured model fitting and switching steps in our algorithm, it will remain 20-40 times faster than the DDMCMC[27].

C. Experiment II: comparison with Graph Cuts and Belief Propagation for stereo

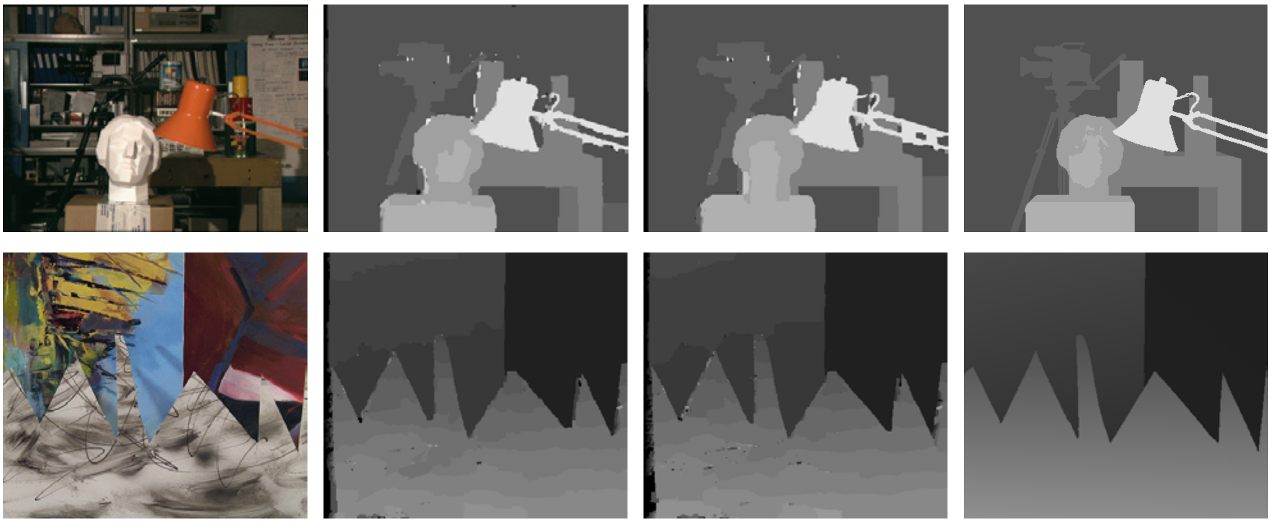
In this section we compare the performance of the SW Cuts with Graph Cuts[4] and Loopy Belief Propagation[33] on stereo matching using the benchmark in [23], [26].

Given a pair of stereo images $\mathbf{I} = (\mathbf{I}_l, \mathbf{I}_r)$, we assign an integer disparity value (as color) $\mathbf{c}_v = d_v$ for every pixel v in the left image. The adjacency graph G_o is simply the lattice with 4-nearest neighbor connections. The energy used in the benchmark[23], [26] is a Potts model with external field,

$$\mathcal{E} = \sum_v D(d_v, v) + \sum_{<s,t>} \beta_{s,t}\mathbf{1}(d_s \neq d_t) \tag{28}$$

The external field (data) term measures the goodness of intensity match between the left and right images for a disparity d_v,

$$D(d_v, v) = \min\{\min_{d_v - 1/2 \leq x \leq d_v + 1/2} |\mathbf{I}_l(v) - \mathbf{I}_r(v - x)|, 50\} \tag{29}$$



a. Left image b. SWC-2 result c. Graph cuts result d. Manual (truth)

Fig. 10. Stereo matching for the Tsukuba sequence (first row) and the Sawtooth sequence (second row).

The coefficient in the prior term is made to be dependent on $< s, t >$ (inhomogeneous Potts model) $\beta_{s,t} = 20$ if $|\mathbf{I}_l(s) - \mathbf{I}_l(t)| > 8$, otherwise $\beta_{s,t} = 40$. This energy has some shortcomings. (i). It is low level Markov random field without generative model fitting.

For example, the slanted planes in Fig. 10 (second row) are broken into many pieces. (ii) It does not treat half-occluded pixels explicitly and because of this, the ground truth have much higher energy than what the algorithms output (see Fig.11). We are forced to use this energy in order to compare with the graph cut (and BP) as this is the type of energy that they can minimize. We compare the SWC-2 with the Graph Cuts implementations provided in the Scharstein and Szelisky's package [23] and Tappen's extension to Belief Propagation [26] available online.

For the stereo problem, we define discriminative probabilities on both vertices and edges to get better empirical results.

On each vertex (pixel) $v \in V$ we compute the vertex probability $q(d_v, v) \propto e^{-D(d_v,v)}$ normalized to 1 for $d_v \in \{0, ..., d_{\max}\}$. It measures how likely pixel v has disparity d_v based on local information. We compute a marginal probability $q(d) = \frac{1}{|V|}\sum_v q(d,v)$ for each disparity level d.

For each edge $e =< s, t >$, we define an edge probability for any $d \in \{0, ..., d_{\max}\}$,

$$q_e^d = 1 - e^{-\frac{20\beta_{s,t}}{3(D(s,d_s)+D(t,d_t))+10}}. \tag{30}$$

Thus we have $d_{\max} + 1$ probabilities on each edge e, one for each disparity level. At each SWC-2 step, we first choose a disparity level d with probability $q(d)$, and then we use q_e^d as the edge probability for clustering the connected component V_o.

We found that most of the energy costs are contributed by the boundary pixels (due to the lack of half-occlusion treatment). Therefore, in SWC-2, a seed vertex v is chosen with equal probability either from the boundary pixels or by sampling from a goodness of fit probability $q(d_v, v)D(d_v, v)$ with d_v being the current assigned disparity at v. That is, we wish to choose more often those pixels v whose assigned disparity level d_v have a lower probability. Then we grow the component V_o as in SWC-2 from the seed v and propose to flip its label. The new disparity level d (or color) for V_o is chosen according to a probability

$$q(d|V_o,\pi) \propto e^{-\sum_{v\in V_o} D(d,v) - 0.7K \sum_{<s,t>, s\in V_o} \beta_{s,t} \mathbf{1}(d\neq d_t)}. \tag{31}$$

Fig. 11 compares the energy curves against CPU time in seconds for the SWC (two runs with different annealing schedules), graph cuts[4], and belief propagation (two versions)[23], [26] We initialized the system with an SWC-1 version working on atomic regions which decreased the energy from about $5,000,000$ to about $650,000$ in less than 30 seconds. Then the SWC-2 version working on the pixel lattice provided the final result. The final energy obtained with SWC-2 was within 1% of the final energy of the Graph Cuts algorithm for the Tsukuba sequence and within less then 2% for the other sequences. All parameters were kept the same in all experiments.

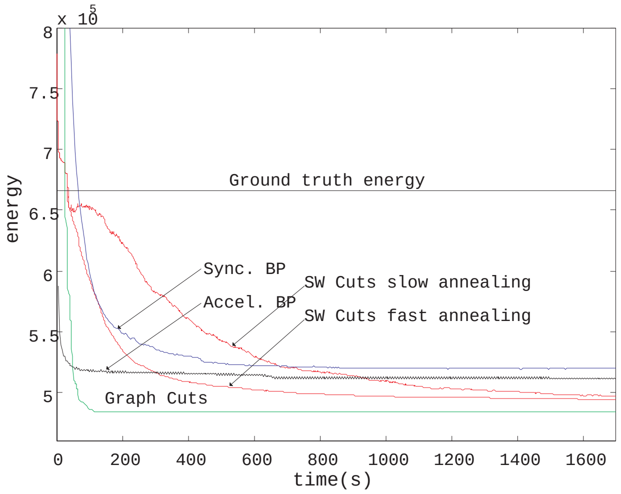
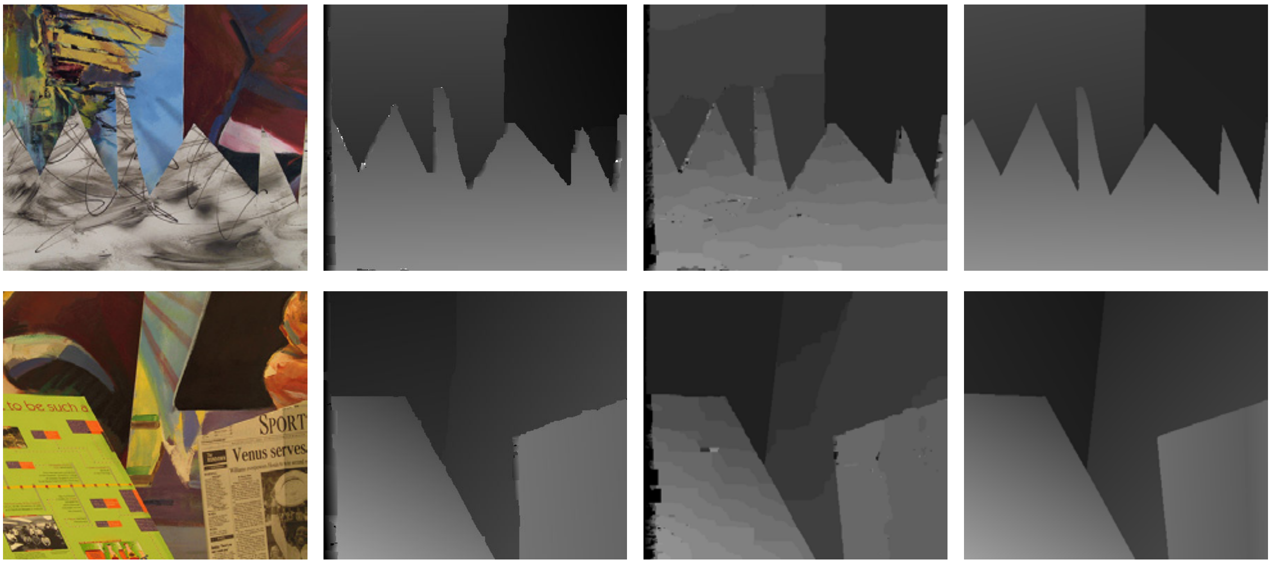


Fig. 11. Performance comparison of SWC with Graph Cuts and Belief Propagation for the Tsukuba sequence. The curves plot the energy over CPU time in seconds.

The energy level is not a good indicator of the quality of results as the ground truth results have higher energy than all algorithms. The experiments show that the SWC reaches lower energy than belief propagation but it is slower than Graph cuts.

If we release ourselves from the simple energy model in eqn.(28), and adopt generative models with piecewise planar surfaces, we obtain a Bayesian posterior probability similar



a. Left image b. SWC result c. Graph cuts d. Manual (truth)

Fig. 12. Using a Bayesian formulation with generative models fitting piecewise planar surfaces, our algorithm obtains much better results for the same set of stereo images. The running time is also reduced to 5 minutes in a PC.

to the segmentation problem using in experiment I. Our algorithm runs in 5 minutes and obtains the much better results shown in Figure 12 which are closer to the ground truth. We run the SWC-2 algorithm on the atomic regions and then run the boundary diffusion[34] for a few steps to smooth the object boundary.

VI. Discussion

In this paper, we present a generic inference algorithm for sampling arbitrary probabilities or energy functions on general graphs by extending the SW method from physics and the Gibbs sampler (SWC-3). Our method extends the SW method from the Metropolis-Hastings perspective and it is thus different from other interpretations in the literature[6], [12]. In fact, there were some early attempts for applying SW to image analysis[12], [3] using a partial decoupling concept.

The speed of the SW-cut method depends on the discriminative probabilities on the edges and vertices. Such probabilities also make a theoretical analysis of convergence difficult. In ongoing projects, we are studying ways for bounding the SW-cut convergence with "external field" (data) and for diagnosing exact sampling using recent advanced techniques. We are also incorporating the SW-Cuts into the DDMCMC framework for image parsing.

Acknowledgment

The work is supported by NSF grant IIS-02-44763 and ONR grant N-00014-02-1-0952. The authors thank Zhuowen Tu and Yingnian Wu for discussions, and they thank Dr. Rangarajan and anonymous reviewers for helpful comments.

References

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

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

[3] S. A. Barker, A. C. Kokaram, and P. J. Rayner. "Unsupervised segmentation of images", *SPIE Conf. on Bayesian Inference for Inverse Problems*, pp.200-211, July 1998.

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

[5] C. Cooper, A. Frieze. "Mixing properties of the Swendsen-Wang process in classes of graphs", *Random Structures and Algorithms* vol. 15, no. 3-4, pp. 242-261, 1999.

[6] R.G. Edwards, A.D. Sokal, "Generalization of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm", *Phys. Rev. Lett.*, 38, pp 2009-2012,1988

[7] C. Fox, G.K. Nicholls, "Exact MAP States and Expectations from Perfect Sampling",*Proc. of 20th Int'l Workshop on Bayesian Inference and Maximum Entropy Methods*, France, 2000.

[8] Y. Gdalyahu, D. Weinshall, M. Werman, "Self-organization in vision: stochastic clustering for image segmentation, perceptual grouping and image database", *IEEE. Trans. on PAMI*, vol. 23, no.10, pp.1053-1074, 2001.

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

[10] V. Gore and A. Sinclair, "The Swendsen-Wang process does not always mix rapidly", *J. Stat. Phys.*, 97(1-2):67-86, 1999.

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

[12] D. Higdon, "Auxiliary variable methods for Markov chain Monte Carlo simulations", *preprint of the Inst. of Stat. and Decision Science*, 1996,

[13] T. Hofmann, J.M. Buhmann, "Pairwise data clustering by deterministic annealing", *IEEE Trans. on PAMI*, vol. 19, no. 1, pp. 1-14, 1997.

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

[15] A.K. Jain, R. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[16] S. Kirkpatrick, C. Gelatt, M. Vecchi, "Optimization by simulated annealing", *Science*, 220(4598), pp.671-680, 1983.

[17] V. Kolmogorov, R. Zabih, "What energy functions can be minimized via graph cuts?", *Proc. ECCV*, pp. 65-81. vol. 3, Copenhagen, Denmark, 2002.

[18] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller,"Equations of the state calculations by fast computing machines", *J. Chem. Physics*, 21, pp.1087-1091,1953

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

[20] J.G. Propp, D.B. Wilson. "Exact Sampling with Coupled Markov Chains and Applications to Statistical Mechanics" *Random Structures and Algorithms*, vol. 9 no. 1-2, pp. 223-252, 1996.

[21] J. Puzicha, T. Hofmann, J.M. Buhmann, "A theory of proximity based clustering: structure detection by Optimization", *Pattern Recognition*, **33**, no.4, pp.617-634, 1999.

[22] S. Roy, I. Cox, "A maximum-flow formulation of the n-camera stereo correspondence problem", *Proc. ICCV*, Bombay, India, 1998.

[23] D. Scharstein and R. Szeliski. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", *IJCV*, 2002

[24] J. Shi, J. Malik, "Normalized cuts and image segmentation", *IEEE Transactions on PAMI*, **22** no 8, pp. 888-905, 2000.

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

[26] M. F. Tappen and W. T. Freeman, "Comparison of Graph Cuts with Belief Propagation for Stereo, using Identical MRF Parameters" *ICCV* 2003

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

[28] Z.W. Tu and S.C. Zhu, "Parsing images into region, curves, and curve processes", *Submitting to IJCV*, Short version appeared in *Proc. of ECCV*, 2002.

[29] J. Wang, et al. "Relationship between ventral stream for object recognition and dorsal stream for spatial vision: an fMRI and ERP study", *Human Brain Mapping*, 8, pp.170-181, 1999.

[30] G. Winkler, *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*, page 171-173, Springer-Verlag, 1995.

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

[32] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: theory and application to image segmentation", *IEEE Trans. on PAMI*, **15**, pp.1101-1113, 1993.

[33] J.S. Yedidia, W.T. Freeman, and Y. Weiss, "Generalized belief propagation", *MERL Report TR-2000-26*, 2000.

[34] S.C. Zhu and A.L. Yuille, "Region competition: unifying snake/balloon, region growing and Bayes/MDL/energy for multi-band image segmentation", *IEEE Trans. on PAMI*, **18**, no. 9, pp.884-900, 1996.

Apendix A. Proof of theorem 1

Although there is combinatorial number of ways for selecting V_o in the two partitions π and π', the proposal probabilities ratio $\frac{q(V_o|\pi)}{q(V_o|\pi'}$ is very simple due to cancellation. In what follows, we compute this ratio for SWC-1. The same ratio can be derived for SWC-2 and SWC-3 following the same steps.

Firstly, we calculate the probability $q(V_o|\pi)$ for selecting V_o in a partition $\pi = (V_1, V_2, ..., V_n)$. Without loss of generality, we assume $V_o \subseteq V_\ell$. At π, the edges between different colors are removed and the set of remaining edges is denoted by

$$E_{\text{on}}(\pi) = E_o \setminus E_{\text{off}}(\pi), \quad E_{\text{off}}(\pi) = \cup_{i \neq j} \mathcal{C}(V_i, V_j). \tag{32}$$

Each edge $e \in E_{\text{on}}(\pi)$ is turned off ($\mu_e = \text{off}$) with probability $1 - q_e$ independently, and we denote the edge variables by

$$U(\pi) = U_{\text{on}}(\pi) \cap U_{\text{off}}(\pi), \quad \text{with } U_{\text{on}}(\pi) = \{\mu_e = \text{on}, e \in E_{\text{on}}(\pi)\},\ U_{\text{off}} = \{\mu_e = \text{off}, e \in E_{\text{on}}(\pi)\}. \tag{33}$$

We denote the sets of edges that are turned on and off by U respectively,

$$E_{\text{on}}(\pi, U) = \{e : e \in E_{\text{on}}(\pi),\ \mu_e = \text{on}\}, \text{ and } E_{\text{off}}(\pi, U) = \{e : e \in E_{\text{on}}(\pi),\ \mu_e = \text{off}\}. \tag{34}$$

The probability of an event $U(\pi)$ is simply

$$p(U(\pi)) = \prod_{e \in E_{\text{on}}(pi,U)} q_e \cdot \prod_{e \in E_{\text{off}}(\pi,U)} (1 - q_e). \tag{35}$$

Then each color V_i is broken into a number n_i of connected components. For event $U = U(\pi)$, we denote the set of connected components by $CP(\pi, U)$. Each set of connected components $CP(\pi)$ can be obtained by a combinatorial number of edge probabilities U, so the probability of $CP(\pi)$ is,

$$p(CP(\pi)) = \sum_{U:CP(\pi,U)=CP(\pi)} p(U(\pi)). \tag{36}$$

We are interested in the set of $CP(\pi)$'s which include V_o as a connected components,

$$\Omega(V_o, \pi) = \{CP(\pi) : V_o \in CP(\pi)\}. \tag{37}$$

Therefore the probability for choosing V_o at π is

$$q(V_o|\pi) = \sum_{CP(\pi,U)\in\Omega(V_o,\pi)} p(U(\pi))p(V_o|CP(\pi, U)). \tag{38}$$

where $p(V_o|CP(\pi, U))$ could be arbitrary, say $p(V_o|CP(\pi, U) = \frac{1}{|CP(\pi,U)|}$.

To summarize, all CPs in $\Omega(V_o, \pi)$ must observe one common property – the edges in the SW-cut $\mathcal{C}(V_o, V_\ell \setminus V_o)$ must be turned off, otherwise V_o is connected to other vertices in V_ℓ and thus violate the premise that V_o is a connected component. So we have

$$\mathcal{C}(V_o, V_\ell \setminus V_o) \subset E_{\text{off}}(\pi, U), \quad \forall CP(\pi, U) \in \Omega(V_o, \pi). \tag{39}$$

Let

$$E^{-}_{\text{off}}(\pi, U) = E_{\text{off}}(\pi, U) \backslash \mathcal{C}(V_o, V_\ell \setminus V_o), \quad \forall CP(\pi, U) \in \Omega(V_o, \pi). \tag{40}$$

Therefore we can take the common factor out the summation,

$$q(V_o|\pi) = \prod_{e\in\mathcal{C}(V_o,V_\ell\setminus V_o)} (1-q_e) \cdot \sum_{CP(\pi,U)\in\Omega(V_o,\pi)} \frac{1}{|CP(\pi, U)|}[\prod_{e\in E_{\text{on}}(pi,U)} q_e \cdot \prod_{e\in E^{-}_{\text{off}}(\pi,U)} (1-q_e)] \tag{41}$$

Secondly, we calculate the probability $q(V_o|\pi')$ for selecting V_o in a partition π'. Without loss of generality, we assume $V_o \subseteq V_{\ell'}$. Following the same steps above, we have,

$$q(V_o|\pi') = \prod_{e\in\mathcal{C}(V_o,V_{\ell'}\backslash V_o)} (1-q_e)\cdot \sum_{CP(\pi',U')\in\Omega(V_o,\pi')} \frac{1}{|CP(\pi',U')|}[\prod_{e\in E_{\text{on}}(pi',U')} q_e \cdot \prod_{e\in E^-_{\text{off}}(\pi',U')} (1-q_e)] \tag{42}$$

Since π and π' are partitions at consecutive SWC-steps, and they differ only in the coloring of V_o, we have the following observations.

For each $CP(\pi,U)\in\Omega(V_o,\pi)$, there is a corresponding $CP(\pi',U')\in\Omega(V_o,\pi')$, such that $CP(\pi',U') = CP(\pi,U)$. Furthermore, we have

$$E_{\text{on}}(\pi,U) = E_{\text{on}}(\pi',U'), \text{ and } E^-_{\text{off}}(\pi,U) = E^-_{\text{off}}(\pi',U'). \tag{43}$$

That is, U and U' differs only in the SW-cuts. As the correspondence is one-to-one, we have

$$\Omega(V_o,\pi) = \Omega(V_o,\pi') \tag{44}$$

Therefore we obtain the ratio by canceling the common probability in eqns.(41) and (42).

$$\frac{q(V_o|\pi)}{q(V_o|\pi')} = \frac{\prod_{e\in\mathcal{C}(V_o,V_\ell\backslash V_o)}(1-q_e)}{\prod_{e\in\mathcal{C}(V_o,V_{\ell'}\backslash V_o)}(1-q_e)}. \tag{45}$$

In a special case, when $q_e = q_o$, $\forall e\in E_o$, we obtain the proposal ratio in equation (9) for the original SW method.

Apendix B. Proof of theorem 2: the splitting and merging cases

For the re-grouping case where $V_o\subset V_\ell$ in π and $V_o\subset V_{\ell'}$ in π', the only way for moving between π and π' is to select V_o. But for the merging and splitting cases there might be two paths illustrated in Fig. 13. Without loss of generality, we write $\pi = (V_1, V_2, V_3, ..., V_n)$ and $\pi' = (V_{1+2}, V_3, V_4, ..., V_n)$ with $V_{1+2} = V1\cup V_2$. The two paths for moving between π and π' are respectively.

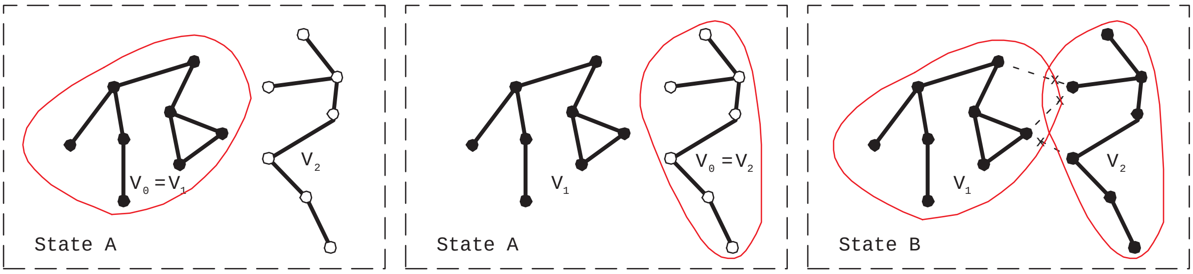


Fig. 13. State π_A has two subgraphs V_1 and V_2 which are merged in state π_B. There are two paths between π_A and π_B. One is to choose $V_0 = V_1$ and the other is to choose $V_0 = V_2$.

Path 1. Choose $V_o = V_1$. In state $\pi = \pi_A$, choose $\ell' = 2$, i.e. merge V_o to V_2, and reversely in state $\pi' = \pi_B$, choose $\ell' = 1$, i.e. split V_o from V_2.

Path 2. Choose $V_o = V_2$. In state $\pi = \pi_A$, choose $\ell' = 1$, i.e. merge V_o to V_1, and reversely in state $\pi' = \pi_B$, choose $\ell' = 2$, i.e. split V_o from V_1.

The proposal probability ratio is,

$$\frac{q(\pi' \to \pi)}{q(\pi \to \pi')} = \frac{q(V_o = V_1|\pi')q(\mathbf{c}_{V_o} = 2|V_o, \pi') + q(V_o = V_2|\pi')q(\mathbf{c}_{V_o} = 1|V_o, \pi')}{q(V_o = V_1|\pi)q(\mathbf{c}_{V_o} = 1|V_o, \pi)) + q(V_o = V_2|\pi)q(\mathbf{c}_{V_o} = 2|V_o, \pi)}. \tag{46}$$

In state $\pi = \pi_A$, the SW-cut $\mathcal{C}(V_o, V_\ell \setminus V_o) = \emptyset$ for both paths, and in state $\pi' = \pi_B$ the cut is $\mathcal{C}(V_\ell, V_{\ell'}) = \mathcal{C}(V_1, V_2)$ for both paths. Following theorem 1, the probability ratios for choosing $V_o = V_1$ and $V_o = V_2$ are equal,

$$\frac{q(V_o = V_1|\pi)}{q(V_o = V_1|\pi')} = \frac{1}{\prod_{e \in \mathcal{C}(V_1, V_2)}(1 - q(e))} = \frac{q(V_o = V_2|\pi)}{q(V_o = V_2|\pi')}. \tag{47}$$

Once V_o is selected, either $V_o = V_1$ or $V_o = V_2$, then the remaining partition for both π and π' are the same, and is denoted by $\pi(V \setminus V_o) = \pi'(V \setminus V_o)$. In proposing the new label of V_o, we easily observe that

$$\frac{q(\mathbf{c}_{V_o} = 2|V_o = V_1, \pi')}{q(\mathbf{c}_{V_o} = 1|V_o = V_2, \pi)} = \frac{q(\mathbf{c}_{V_o} = 1|V_o = V_2, \pi')}{q(\mathbf{c}_{V_o} = 2|V_o = V_1, \pi)}. \tag{48}$$

Then the acceptance rate in theorem 2 follows from equations (47) and (48).