Graph Partition by Swendsen-Wang Cuts

Adrian Barbu and Song-Chun Zhu
University of California, Los Angeles
Departments of Computer Science and Statistics
abarbu@ucla.edu, sczhu@stat.ucla.edu

Abstract

Vision tasks, such as segmentation, grouping, recognition, can be formulated as graph partition problems. The recent literature witnessed two popular graph cut algorithms: the Ncut using spectral graph analysis and the minimum-cut using the maximum flow algorithm. This paper presents a third major approach by generalizing the Swendsen-Wang method– a well celebrated algorithm in statistical mechanics. Our algorithm simulates ergodic, reversible Markov chain jumps in the space of graph partitions to sample a posterior probability. At each step, the algorithm splits, merges, or re-groups a sizable subgraph, and achieves fast mixing at low temperature enabling a fast annealing procedure. Experiments show it converges in 2-30 seconds in a PC for image segmentation. This is 400 times faster than the single-site update Gibbs sampler, and 20-40 times faster than the DDMCMC algorithm. The algorithm can optimize over the number of models and works for general forms of posterior probabilities, so it is more general than the existing graph cut approaches.

1. Introduction

Computer vision problems, such as image segmentation, perceptual organization, and object recognition, require grouping image elements (pixels, edgelets, primitives) into "coherent" visual patterns (regions, curves, objects) in a process of optimizing some grouping criteria. The problem can be represented in an adjacency graph with the vertices being the image elements, the edges being spatial relationships and subgraphs being coherent visual patterns. Thus it becomes a graph partition problem.

There are two approaches for graph partition in the recent literature. One is the normalized cut[12, 9] using graph spectral analysis to optimize a discriminative criterion. The other is the minimum-cut[8, 6] which maps an energy minimization problem to a maximum flow algorithm. The latter is solved in polynomial time. Despite their reasonable success, the two approaches are far from being general solutions. Firstly it was shown[6] that only very limited classes of energy functions can be mapped to the maximum flow problem. Secondly the graph spectral analysis, like many other discriminative clustering algorithms[3, 2], has difficulties in expressing global visual patterns, such as shading effects, perspective projection effects, contour closure etc. Furthermore natural images contain very diverse visual patterns which are "coherent" in many different ways. This requires a generative and Bayesian formulation incorporating a number of diverse and competing image models[10]. There is no single discriminative criterion that is generally applicable to all the visual patterns.

In this paper, we present a third major graph partition approach by generalizing the Swendsen-Wang method– a well celebrated algorithm in statistical mechanics. Formulated in a Bayesian framework with generative image models, our algorithm simulates ergodic and reversible Markov chain jumps in the space of all possible graph partitions to search for global optima. The basic ideas and contributions of our method are:

1. Given an adjacency graph, we compute a local probability at each edge for how likely the two vertices (image elements) belong to the same pattern. Then by turning on the edges at random according to their associated probabilities, we form connected components, each being a good candidate for a coherent pattern.

2. At each step, the algorithm splits, merges, or regroups a connected component which often includes a big number of vertices. The moves are ergodic and observe detailed balance equations. The candidate states are selected proportional to their posterior probabilities weighted by the probabilities of "graph cuts". The acceptance probability can be made to be always one, and thus our algorithm becomes a generalized Gibbs sampler.

3. The algorithm "mixes" rapidly at low temperature. Unlike most MCMC methods it no longer needs a long simulated annealing procedure[5]. Instead a fast annealing, starting from a lower temperature is used. Thus it can start from good initial conditions using heuristics to achieve a short "burn-in" period. As a result, the algorithm is about

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 **IEEE**

IEEE COMPUTER SOCIETY

400 times faster than the classical Gibbs sampler[4] which flips a single vertex each time, and it is 20-40 times faster than the previous DDMCMC algorithm[10]. It converges in 2-30 seconds in a 1.5GHz PC for image segmentation.

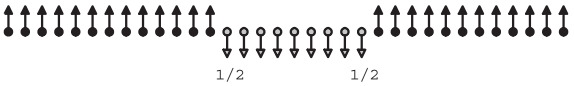
The central contribution of the paper is the two mathematical theorems for calculating the acceptance probabilities for the big moves, which observe miraculous cancellations in the calculation. The theorems ensure that our algorithm samples from general posterior probabilities, and provide a foundation for fast simulation and optimization for a broad range of vision problems.

2. Swendsen-Wang: basic ideas

The difficulty of sampling the graph partition space is well reflected in the Ising and Potts models in statistical mechanics,

$$p(\mathbf{I}) \propto \exp\{\beta \sum_{<s,t>} \mathbf{1}(\mathbf{I}_s = \mathbf{I}_t)\} \quad \beta > 0. \qquad (1)$$

where $\mathbf{1}(\mathbf{I}_s = \mathbf{I}_t) = 1$ if $\mathbf{I}_s = \mathbf{I}_t$ for adjacent spins s, t otherwise it is zero.



For the string of spins with label $\mathbf{I} \in \{\pm 1\}^n$ shown above, the highest probability is achieved when all vertices have the same label. In a best visiting scheme, the Gibbs sampler flips the -1 spins at the two "cracks" to $+1$ with probability $p_o = 1/2$. Thus to flip a string of n spins ($n = 9$ here) from -1 to $+1$, the expected number of steps is $\frac{1}{(1/p_o)^n} = 2^n$. This is exponential waiting!

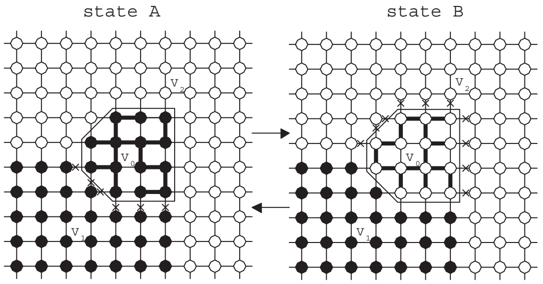


Figure 1. The Swendsen-Wang algorithm flips a patch of spins in one step.

A major speedup for the Ising model in equation (1) is achieved by the Swendsen-Wang (1987) algorithm [11]. E.g. Fig. 1 shows an adjacency graph as a 2D lattice with each edge e connecting two adjacent spins s, t. SW turns "on" each edge e with a constant probability $q_o = 1 - e^{-\beta}$ if s, t have the same label. Fig. 1 shows a component V_0 connected by bold edges which are turned "on" at two states A and B. The edges between V_0 and its neighbors – V_1 in state A and V_2 in state B are cut – turned "off", see the crosses in the figure. We denote the two sets of edges by the respective "cuts"

$$\mathcal{C}(V_0, V_1), \quad \mathcal{C}(V_0, V_2).$$

Then SW flips all spins in V_0 in a single step and makes a reversible jump between states A and B. The acceptance probability for the move is shown to be 1. So SW can flip all -1 spins in the 1D string example in one or a few steps.

The SW algorithm achieves fast mixing even at critical temperature for typical graphs. Unfortunately, it is limited to simple Ising/Potts models and does not use the image (data) information in forming the component V_0. It is found to be ineffective in the presence of external field (data). In the following, we extend SW to simulating general Bayesian posterior probabilities and make use of bottom-up information to form the candidate components V_0 to further speed up the computation.

3. Bayesian formulation of graph partition

3.1. Graph Partition

Let $G_o =< V, E_o >$ be an adjacency graph where $V = \{v_1, v_2, ..., v_N\}$ is the set of vertices for image elements such as pixels, edgelets, primitives and E_o is a set of edges $e =< s, t >$ for adjacent elements s, t. The objective is to partition graph G_o into unknown number of n *full subgraphs* $G_k =< V_k, E_k >, k = 1, 2, ..., n$, each keeping all the edges in G_o that connect its vertices:

$$V = \cup_{k=1}^{n} V_k, \quad V_k \neq \emptyset,\ V_i \cap V_j = \emptyset \text{ for } i \neq j.$$
$$E_k = \{e = (u, v) \in E_o \mid u, v \in V_k\}, \quad k = 1, 2, ..., n.$$

We denote by π_n a partition with n subgraphs.

$$\pi_n = \{V_1, V_2, ..., V_n\} \text{ or } \{G_1, G_2, ..., G_n\}$$

Vertices in each subset $V_k, k = 1, 2..., n$ forms a coherent visual pattern specified by a generative probability.

The space of all possible partition is denoted by

$$\Omega_\pi = \cup_{n=1}^{|V|} \Omega_{\pi_n}$$

with Ω_{π_n} being the space of all n-partitions. The edges between any two sets V_i and V_j are denoted by a *cut*

$$C(V_i, V_j) = \{e =< s, t >: e \in E_o, s \in V_i, t \in V_j\}, \quad i \neq j.$$

Fig.2 shows a typical example of image segmentation. We obtain an over-segmentation (middle) by applying a Canny

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

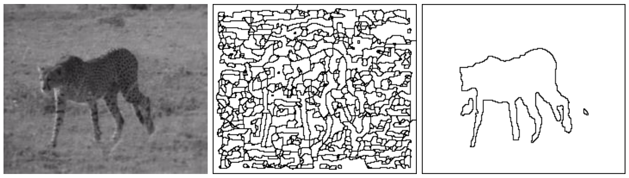


Figure 2. Input image, an over-segmentation with "atomic" regions being vertices, and a segmentation result.

edge detection followed by edge tracing to form "atomic" regions with nearly constant intensities. These atomic regions are the vertices V_o and any two adjacent atomic regions are connected by an edge to form the graph G_o. The right image is a result of our partition algorithm.

3.2. Solution space and Markov chain design

Take image segmentation as an example, we denote by $\mathbf{I}_v$ the observed image attributes (pixel intensity, edge position and orientation, etc) for element v, and by $\mathbf{I}_V$ the image representation for the set V. Suppose we use L classes of image models for various patterns, such as color, texture, shading, curve etc. Each type of model is indexed by $c \in \{C_1, C_2, ..., C_L\} = \Omega_C$, and specified with parameters $\theta_i \in \Omega_{c_i}$. The model space is the union

$$\Omega_\theta = \cup_{c \in \Omega_C} \Omega_\ell.$$

The inner representation for a segmentation is

$$W = (n, \pi_n, (c_1, \theta_1), (c_2, \theta_2), ..., (c_n, \theta_n)) \qquad (2)$$

Each subgraph $V_i, i = 1, 2, ..., n$ in partition π_n is specified by a model $p(\mathbf{I}_{V_i}; c_i, \theta_{c_i})$ of type c_i and parameters θ_i.

The solution space for W is

$$\Omega = \cup_{n=1}^{N} \{\Omega_{\pi_n} \times \Omega_C^n \times \Omega_{c_1} \times \cdots \times \Omega_{c_n}\}.$$

This factorization of the solution space corresponds to the necessary solution steps:

1. Partition graph G_o by finding $\pi_n \in \Omega_\pi$.

2. Select an image model $c \in \Omega_C$ for each subgraph $V_i \in \pi_n$.

3. Fit the models $\theta_{c_i} \in \Omega_{c_i}, i = 1, 2, ..., n$.

If we assume the patterns are mutually independent, then the objective is to simulate a Bayesian posterior

$$W \sim p(W|\mathbf{I}) \propto \prod_{i=1}^{n} p(\mathbf{I}_{V_i}; c_i, \theta_{c_i}) p(W). \qquad (3)$$

The prior and image models can be Markov random field models or global spline models, and are beyond what can be minimized by the graph cut algorithms[6, 9].

The Markov chain must be ergodic in space Ω and have $p(W|\mathbf{I})$ as its stationary probability. In short, we need two types of reversible jumps[1] bridging the subspaces of different dimensions in Ω.

1. Jumps in the model space $\Omega_C^n \times \Omega_{c_1} \times \cdots \times \Omega_{c_n}$, such as model switching, diffusion (fitting) of parameters θ_c.

2. Jumps in the partition space Ω_π: split, merge, death, birth.

The jumps are realized by Metropolis-Hastings methods[7]. For a pair of states $W = A$ and $W = B$, we need to design *proposal probabilities* $q(A \to B)$ and $q(B \to A)$. The recent idea of data-driven Markov chain Monte Carlo (DDMCMC) in [10] is to calculate them based on bottom-up discriminative models, summarized by $D(\mathbf{I})$, so that the proposal probabilities approximate the posterior

$$q(B \to A) = q(A|B, D(\mathbf{I})) \approx p(A|\mathbf{I})$$
$$q(A \to B) = q(B|A, D(\mathbf{I})) \approx p(B|\mathbf{I})$$

Then the proposed move from A to B is accepted with high probability $\alpha(A \to B)$

$$\alpha(A \to B) = \min(1, \frac{q(A|B, D(\mathbf{I}))}{q(B|A, D(\mathbf{I}))} \cdot \frac{p(B|\mathbf{I})}{p(A|\mathbf{I})}). \qquad (4)$$

In this paper we use bottom-up data-driven information $D(\mathbf{I})$ and go one step further by making big moves. So the algorithm can reach from a state A to very different B in one step which may need an exponential number of small moves otherwise, as we discussed in the Ising model example.

We follow the DDMCMC method[10] for the jumps in model space, and the rest of the paper is focused on designing smart moves in the partition space Ω_π for fast convergence and mixing.

4. Sampling Ω_π with discriminative models

For an adjacency graph $G_o =< V, E_o >$, we augment each edge $e =< s, t > \in E_o$ with a binary random variable $\mu_e \in \{\text{on}, \text{off}\}$ representing whether the edge is turned "on" or "off". In contrast to a constant probability q_o for all edges in the SW algorithm, we compute a discriminative model for $q_e = q(\mu_e = \text{on}|F(s), F(t))$ based on local vector valued features (texture, color, geometry etc.) $F(s), F(v)$ at the two sites. q_e indicates how coherent (or similar) the two vertices s and t are, and can be trained off-line.

By turning "on" each edge e in G_o with probability q_e independently, we obtain a sparse graph $G =< V, E >$ with $E \subset E_o$ being the set of edges which are turned on by chance. The probability for E or G is

$$q(E) = \prod_{e \in E} q_e \prod_{e \in E_o - E} (1 - q_e). \qquad (5)$$

$G =< V, E >$ consisting of a number n of connected components $g_k =< V_k, E_k >$.

$$G = \cup_{k=1}^{n} g_k,\ \cup_{k=1}^{n} V_k = V,\ \cup_{k=1}^{n} E_k = E.$$

We denote them by

$$CP = \{V_1, V_2, ..., V_n\}. \qquad (6)$$

As the local probabilities q_e are well trained, the subgraphs in CP are often meaningful parts of patterns. This way, $q(E)$ defines a bottom-up probability $q(\pi)$ on the partition space Ω_π.

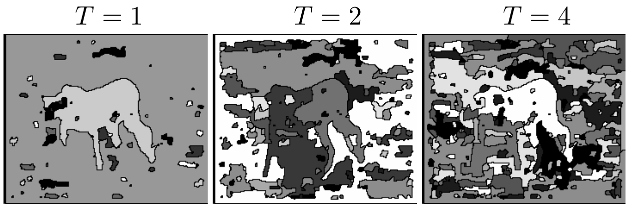
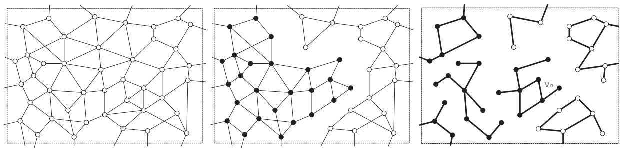


Figure 3. Random samples of CP at $T = 1, 2, 4$ according to $q(E)$.

Figure 3 shows random graph partitions CP for the cheetah image whose adjacency graph G_o is built on the atomic regions in Fig. 2 (middle). On each column, we show a CP sampled according to $q(E)$ in equation (5). The size of the components of CP can be controlled by a temperature T on the edge probabilities q_e^T. The smaller the T, the larger the size of the components. Clearly various parts of the cheetah are obtained, which will be used as candidates for big and meaningful moves in our MCMC algorithm.

5. Stochastic graph partition by MCMC



a. G_o b. G c. CP

Figure 4. Three stages of graphs: a. adjacency graph G_o, b. current partition state G, c. a sample from the discriminative models of G and its connected components CP.

Our graph partition algorithm operates three graphs shown in Figure 4. It starts with an adjacency graph $G_o =< V, E_o >$. The current Markov chain state is a partition $\pi : V = \cup_{l=1}^{n} V_l$, represented by a graph $G = \cup_{l=1}^{n} G_l$, where $G_l =< V_l, E_l >, l = 1, 2, ..., n$ are *full subgraphs* of G_o (Fig.4.b), i.e. G was obtained from G_o by removing the edges between the subsets V_l.

Then during a move between two partition states, it generates connected components CP (Fig.4.c) by turning on/off the edges in G. A component in CP is picked up at random as a candidate for reassignment.

```
Swendsen-Wang Cuts: SWC-1
Input: $G_o =< V, E_o >$, discriminative probabilities $q_e, \forall e \in E_o$,
    and generative posterior probability $p(W|\mathbf{I})$.
Output: Samples $W \sim p(W|\mathbf{I})$.
1. Initialize a graph partition $\pi$: $G = \cup_{l=1}^{n} G_l$.
2. Repeat, for current state A
3.     Repeat for each subgraph $G_l =< V_l, E_l >, l = 1, 2, ..., n$
4.         For $e \in E_l$, turn $\mu_e =$ on with probability $q_e$.
5.         $V_l$ is divided into $n_l$ connected components:
                   $\{g_{li} =< V_{li}, E_{li} >, i = 1, ..., n_l\}$.
6.     Collect connected components from all subgraphs (see Fig.4.c)
         $CP = \{V_{li} : l = 1, ..., n, i = 1, ..., n_l\}$.
7.     Select a component $V_0 \in CP$ at random with probability
         $q(V_0|CP)$, (usually $1/|CP|$) (see Fig.5.a).
8.     Propose to assign $V_0$ to a subgraph $G_{l'}$. $l'$ follows a probability
         $q(l'|V_0, A, G_o)$ (state $B$ in Fig.5.b if $V_0$ is merged to an
         existing subgraph, state $C$ in Fig.5.c if $V_0$ is a new subgraph).
9.     Accept the move with probability
             $\alpha(A \to B)$ or $\alpha(A \to C)$ in theorem 1.
```

We omit the parallel steps of model switching and fitting for clarity. The probability $q(l'|V_0, A, G_o), l' = 1, ..., n+1$ can be designed simply as follows:

$$q(l'|V_0, A, G_o) = \begin{cases} a & \text{if } G_{l'} \text{ is adjacent to } \mathrm{V}_0, \\ b & \text{if } l' = n+1, \text{ new subgraph} \\ c & \text{else} \end{cases}$$

such that $\sum_{l'=1}^{n+1} q(l'|V_0, A, G_o) = 1$. Usually $a = b = 10c$.

The move between states A and B is a split-merge operation in canonical cases. Two special cases are the birth and death moves.

1. If $l' = n + 1$, V_0 becomes a new subgraph, so the move is a birth operation.

2. If V_0 is equal to a subgraph V_l, the whole subgraph G_l is merged to $G_{l'}$. The number of subgraphs is reduced by one, so it is a death operation.

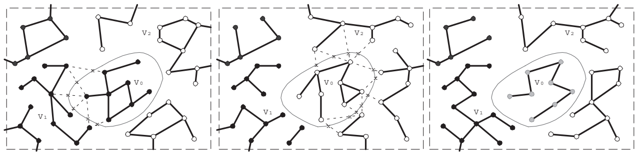
In what follows, we give a simple, explicit expression for the acceptance probability, which can be made to be 1 through a smarter choice of $q(l'|V_0, A, G_o)$.

Theorem 1 *In the above notation, consider a candidate component V_0 selected by* SWC-1. *If the proposed move to reassign V_0 from G_l to $G_{l'}$ is accepted with probability*

$$\alpha(A \to B) = \min(1, \frac{\prod_{e \in \mathcal{C}(V_0, V_{l'} - V_0)} (1 - q_e)}{\prod_{e \in \mathcal{C}(V_0, V_l - V_0)} (1 - q_e)} \frac{q(l|V_0, B, G_o)}{q(l'|V_0, A, G_o)} \frac{p(B|\mathbf{I})}{p(A|\mathbf{I})}) \qquad (7)$$

then the Markov chain is ergodic and observes the detailed balance equations.

In the special case when $l' = n + 1$, V_0 is proposed to be a new subgraph and $V_{l'} - V_0 = \emptyset$. So the cut is empty $\mathcal{C}(V_0, V_{l'} - V_0) = \emptyset$, $\prod_{e \in \mathcal{C}(V_0, V_{l'} - V_0)} (1 - q_e) = 1$ and $\alpha(A \to B)$ becomes $\alpha(A \to C)$.



a. state A b. state B c. state C

Figure 5. A move between partition states $\pi = A, B, C$, different by a set of vertices V_0. The vertices in the same color belong to a subgraph. The vertices connected by thick edges form a connected component.

Proof. The idea of the proof is that even though the proposal probabilities $q(A \to B)$ and $q(B \to A)$ are very complicated, their ratio $q(B \to A)/q(A \to B)$ is extremely simple through miraculous cancellation. Then the conclusion follows from the Metropolis-Hastings equation (4).

First, we calculate the proposal probability $q(A \to B)$ in SWC-1, assuming state A has n subgraphs $G_l =< V_l, E_l >$ $, l = 1, 2, ..., n$. In the canonical case when $V_0 \neq V_l$ and $V_{l'} \neq \emptyset$, it is a conditional probability which consists of two steps: (1) choosing V_0 and (2) choosing l'. For clarity, we discuss the exception cases later.

In state A, each subgraph G_l is broken into connected components CP_l by turning on and off the edges in E_l at random. We denote the set of all connected components

$$CP(A) = \cup_l CP_l = \{V_{li} : l = 1, ..., n;\ i = 1, ..., n_l\}.$$

For example, Figure 5.a shows 6 connected components. For a CP of state A, we denote by $E_{\text{on}}(A, CP)$ the edges that are turned on (the thick edges in Figure 5.a)

$$E_{\text{on}}(A, CP) = \cup_{l=1}^{n} \{\cup_{i=1}^{n_l} E_{ki}\}.$$

The rest of the edges, which are turned off, are the "cuts" between a connected component V_{li} and other vertices in the subgraph, i.e. $V_l - V_{li}$,

$$E_{\text{off}}(A, CP) = \cup_{l=1}^{n} \{\cup_{i=1}^{n_l} C_{li}\}, \quad C_{li} = C(V_{li}, V_l - V_{li}).$$

Note that the edges between subgraphs had been turned off before entering state A. The probability for choosing a CP depends on state A and the discriminative models $D(\mathbf{I})$,

$$q(CP|A, D(\mathbf{I})) = \prod_{e \in E_{\text{on}}(A,CP)} q_e \prod_{e \in E_{\text{off}}(A,CP)} (1 - q_e).$$

We denote by $\Omega_{CP}(A)$ the set of all possible CP's at state A. We are interested in those CP's which contain V_0,

$$\Omega^0_{CP}(A) = \{CP(A) : V_0 \in CP\}.$$

Without loss of generality, we assume that V_0 is a component from subgraph $G_1 =< V_1, E_1 >$. We denote the cut between V_0 and $V_1 - V_0$ by $\mathcal{C}_{01} = C(V_0, V_1 - V_0)$.

All CPs in $\Omega^0_{CP}(A)$ have the following two properties: they all contain V_0, and all edges between V_0 and $V_1 - V_0$ are turned off (otherwise V_0 is connected to other vertices). In other words, $\forall CP \in \Omega^0_{CP}(A)$

$$V_0 \in CP \text{ and } C_{01} \subset E_{\text{off}}(A, CP)..$$

For each $CP \in \Omega^0_{CP}(A)$, the set V_0 is picked with a probability $q(V_0|CP)$. Now we are ready to compute the probability for selecting V_0 at state A,

$$q(V_0|A, D(\mathbf{I})) = \sum_{CP \in \Omega^0_{CP}(A)} q(V_0|CP) q(CP|A, D(\mathbf{I})) \quad (8)$$

$$= \prod_{e \in \mathcal{C}_{01}} (1 - q_e) [\sum_{CP \in \Omega^0_{CP}(A)} q(V_0|CP) \prod_{e \in E_{\text{off}}(A,CP) - \mathcal{C}_{01}} (1 - q_e) \prod_{e \in E_{\text{on}}(A,CP)} q_e].$$

We were able to factor the product $\prod_{e \in C_{01}} (1 - q_e)$ out because $C_{01} \subset E_{\text{off}}(A, CP)$ for all $CP \in \Omega^0_{CP}(A)$.

Once V_0 is selected, it is assigned to $G_{l'}$ with probability $q(l'|V_0, A, G_o)$, the same for all $CP \in \Omega^0_{CP}(A)$. Therefore, the proposal probability from A to B is,

$$q(A \to B) = q(V_0|A, D(\mathbf{I})) q(l'|V_0, A, G_o). \quad (9)$$

Now we calculate the proposal probability $q(B \to A)$ in algorithm SWC-1. In the canonical case, the only way one can get from state B to state A is by selecting V_0 as a connected component and re-assigning it to G_l.

In state B, we have the same partition as in state A except that V_0 belongs to $G_{l'}$ (see Fig. 5.b). Without loss of generality, we assume that V_0 is a component from the subgraph $G_2 =< V_2, E_2 >$. $\Omega^0_{CP}(B)$ is the set of CP's that contain V_0 as a component and must share the common cut $\mathcal{C}_{02} = C(V_0, V_2 - V_0)$, illustrated in Figure 5.b by the crosses. Similarly, the probability for selecting V_0 at state B is,

$$q(V_0|B, D(\mathbf{I})) = \sum_{CP \in \Omega^0_{CP}(B)} q(V_0|CP) q(CP|B, D(\mathbf{I})) \quad (10)$$

$$= \prod_{e \in \mathcal{C}_{02}} (1 - q_e) [\sum_{CP \in \Omega^0_{CP}(B)} q(V_0|CP) \prod_{e \in E_{\text{off}}(B,CP) - \mathcal{C}_{02}} (1 - q_e) \prod_{e \in E_{\text{on}}(B,CP)} q_e],$$

and the proposal probability from B to A is,

$$q(B \to A) = q(V_0|B, D(\mathbf{I})) q(l|V_0, B, G_o). \quad (11)$$

Observation. For each $CP \in \Omega^0_{CP}(A)$, then $CP \in \Omega^0_{CP}(B)$ and vice versa. Therefore we have

$$\Omega^0_{CP}(A) = \Omega^0_{CP}(B) \tag{12}$$

For any CP above, the set of edges turned on are the same,

$$E_{\text{on}}(A, CP) = E_{\text{on}}(B, CP) \tag{13}$$

and the set of edges turned off are also the same except cut $\mathcal{C}_{01}$ occurs in state A and cut $\mathcal{C}_{02}$ occurs in state B. So

$$E_{\text{off}}(A, CP) - \mathcal{C}_{01} = E_{\text{off}}(B, CP) - \mathcal{C}_{02}. \tag{14}$$

Plug in equations (13) and (14) into equations (9) and (11), we have the probability ratio by cancellation,

$$\frac{q(V_0|B, D(\mathbf{I}))}{q(V_0|A, D(\mathbf{I}))} = \frac{\prod_{e\in\mathcal{C}_{02}}(1-q_e)}{\prod_{e\in\mathcal{C}_{01}}(1-q_e)}. \tag{15}$$

Therefore,

$$\frac{q(B\to A)}{q(A\to B)} = \frac{\prod_{e\in\mathcal{C}_{02}}(1-q_e)}{\prod_{e\in\mathcal{C}_{01}}(1-q_e)} \cdot \frac{q(l|V_0, B, G_o)}{q(l'|V_0, A, G_o)}.$$

By equation (4), we obtain $\alpha(A \to B)$ as the theorem states. Thus the move between A and B observes the detailed balance equations.

The above proof is for the canonical case when there is only one way to go from state A to state B, or from state B to state A, namely by reassigning V_0.

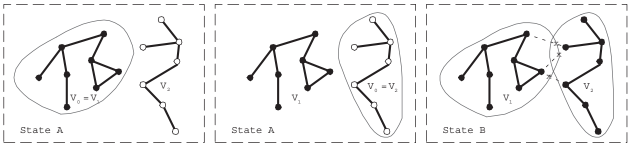


Figure 6. There are two ways to merge subgraphs V_1, V_2 from state A to get to state B. One is to choose V_1 and merge it to V_2, the other is to choose V_2 and merge it to V_1.

There is an exception to the canonical case when there are two paths between states A and B. It occurs when a whole subgraph G_l or $G_{l'}$ is chosen as V_0 in state A, and thus two subgraphs are merged in state B. Without loss of generality, we only consider two subgraphs V_1, V_2 in state A and one subgraph $V_1 \cup V_2$ in state B, as Fig. 6 displays.

- *Path 1*. Choose $V_0 = V_1$. In state A, choose $l' = 2$, i.e. merge it to V_2, and reversely in state B, choose $l' = 1$, i.e. split it from V_2.
- *Path 2*. Choose $V_0 = V_2$. In state A, choose $l' = 1$, i.e. merge it to V_1, and reversely in state B, choose $l' = 2$, i.e. split it from V_1.

Thus the proposal probability $q(A \to B)$ is the sum of the probabilities for the two paths.

$$\begin{aligned} q(A\to B) &= q(l'=2|V_1, A, G_o)q(V_1|A, D(\mathbf{I})) \\ &\quad + q(l'=1|V_2, A, G_o)q(V_2|A, D(\mathbf{I})) \end{aligned} \tag{16}$$

and similarly

$$\begin{aligned} q(B\to A) &= q(l'=1|V_1, B, G_o)q(V_0=V_1|B, D(\mathbf{I})) \\ &\quad + q(l'=2|V_2, B, G_o)q(V_0=V_2|B, D(\mathbf{I})). \end{aligned} \tag{17}$$

In state A, the cut is $\mathcal{C}(V_0, V_l - V_0) = \mathcal{C}(V_0, \emptyset) = \emptyset$ for both paths, and in state B the cut is $\mathcal{C}(V_0, V_l - V_0) = \mathcal{C}(V_1, V_2) = \mathcal{C}_{12}$ for both paths.

Following previous calculation, we have the proposal probability ratio for choosing $V_0 = V_1$ in path 1,

$$\frac{q(V_0=V_1|B, D(\mathbf{I}))}{q(V_0=V_1|A, D(\mathbf{I}))} = \frac{\prod_{e\in\mathcal{C}(V_1,V_2)}(1-q_e)}{\prod_{e\in\mathcal{C}(V_1,\emptyset)}(1-q_e)} = \prod_{e\in\mathcal{C}_{12}}(1-q_e). \tag{18}$$

Similarly, we have the probability ratio for choosing $V_0 = V_2$ in path 2,

$$\frac{q(V_0=V_2|B, D(\mathbf{I}))}{q(V_0=V_2|A, D(\mathbf{I}))} = \frac{\prod_{e\in\mathcal{C}(V_2,V_1)}(1-q_e)}{\prod_{e\in\mathcal{C}(V_2,\emptyset)}(1-q_e)} = \prod_{e\in\mathcal{C}_{12}}(1-q_e). \tag{19}$$

Plug in the above equations, we obtain the ratio,

$$\begin{aligned} \tfrac{q(B\to A)}{q(A\to B)} &= \textstyle\prod_{e\in\mathcal{C}_{12}}(1-q_e) \qquad (20) \\ &\cdot \tfrac{q(l'=1|V_1,B,G_o)q(V_1|A,D(\mathbf{I}))+q(l'=2|V_2,B,G_o)q(V_2|A,D(\mathbf{I}))}{q(l'=2|V_1,A,G_o)q(V_1|A,D(\mathbf{I}))+q(l'=1|V_2,A,G_o)q(V_2|A,D(\mathbf{I}))} \end{aligned}$$

The proposal probabilities for l' must be designed in such a way that:

$$\frac{q(l'=1|V_1, B, G_o)}{q(l'=2|V_1, A, G_o)} = \frac{q(l'=2|V_2, B, G_o)}{q(l'=1|V_2, A, G_o)} \tag{21}$$

This is easily satisfied in general. So we have,

$$\frac{q(B\to A)}{q(A\to B)} = \prod_{e\in\mathcal{C}(V_0, V_{l'}-V_0)}(1-q_e)\cdot\frac{q(l'=1|V_1, B, G_o)}{q(l'=2|V_1, A, G_o)} \tag{22}$$

In general notation, it is

$$\frac{q(B\to A)}{q(A\to B)} = \frac{\prod_{e\in\mathcal{C}(V_0,V_{l'}-V_0)}(1-q_e)}{\prod_{e\in\mathcal{C}(V_0,V_l-V_0)}(1-q_e)} \cdot \frac{q(l|V_0, B, G_o)}{q(l'|V_0, A, G_o)}$$

Thus we have proved the exception case.

To prove ergodicity, observe that there is a non-zero probability that any given node is chosen as a connected component V_0. Since the node can then be assigned to any other subgraph with nonzero probability, and this holds for all nodes independently, we can get from any partition to any other partition with non-zero probability. *End of Proof.*

Now we shall construct $q(l'|V_0, A, G_o)$ in such a way to obtain acceptance probability 1. Then our algorithm becomes a generalized Gibbs sampler.

Suppose the Markov chain is at a partition state $A = (V_1, V_2, ..., V_n)$, and a connected component $V_0 \subset V_l$ is selected by SWC-1 as a candidate set. We have $n+1$ choices for state B by assigning V_0 to one of the following vertex sets:

$$\{S_1 = V_1,\ S_2 = V_2, ..., S_l = V_l - V_0, ...,\ S_n = V_n, S_{n+1} = \emptyset\}$$

We denote the states as $B_1, B_2, ..., B_{n+1}$ respectively. Clearly $B_l = A$ and in state B_{n+1}, V_0 is a new subgraph. In the exception case $V_0 = V_l$, then the state $B_{n+1} = A$ is redundant, so one of them should be eliminated.

Denote the cuts between V_0 and S_j by $\mathcal{C}_j = C(V_0, S_j)$ $j = 1, 2, ..., n+1$ with $\mathcal{C}(V_0, \emptyset) = \emptyset$.

Theorem 2 *In the above notation, suppose V_0 is a candidate vertex set selected by SWC-1, in partition state A. If the probabilities for merging V_0 to $V_{l'}$ are chosen to be*

$$q(l'|V_0, A, G_o) \propto \prod_{e \in \mathcal{C}_{l'}} (1 - q_e) \cdot p(B_{l'} \mid \mathbf{I}). \qquad (23)$$

then the proposed move is accepted with probability $\alpha(A \to B_{l'}) = 1$.

The proof is straightforward and we omit it. We also omit the proof that $q(l'|V_0, A, G_o)$ satisfies condition (21). In practice, the posteriors $p(A \mid \mathbf{I})$ and $p(B_{l'} \mid \mathbf{I})$ only involve local computation and the cuts $\mathcal{C}_{l'}$ are small or empty.

Intuitively, our algorithm samples a random set of vertices according to posterior for goodness-of-fit modulated by the cut probability to achieve detailed balance. This is much more general than the original SW-method [11] and the Gibbs sampler [4].

6. Experiments and performance analysis

The image segmentation experiment was performed on "atomic regions" obtained by edge detection and edge tracing. They form the nodes of our graph. The discriminative probability q_e for an edge $e =< v_i, v_j >$ is

$$q_e = 0.1 + 0.9e^{-(KL(p_i||p_j)+KL(p_j||p_i))/2}. \qquad (24)$$

where p_i, p_j are 15 bin intensity histograms in the atomic regions, and $KL()$ is the Kullback-Leibler divergence. In general, this q_e can be learned through supervised learning. We use three simple image models $\{C_1, C_2, C_3\}$ (constant, linear and quadratic polynomial intensity) with additive noise modeled by a 15 bin histogram $\mathcal{H}$.

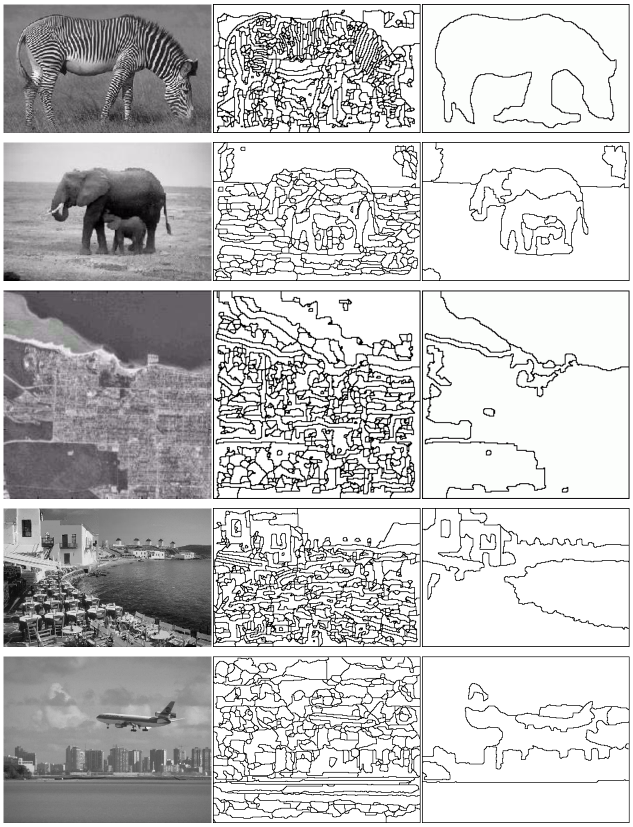


Figure 7. Image segmentation: input image, atomic regions as image elements and the segmentation result.

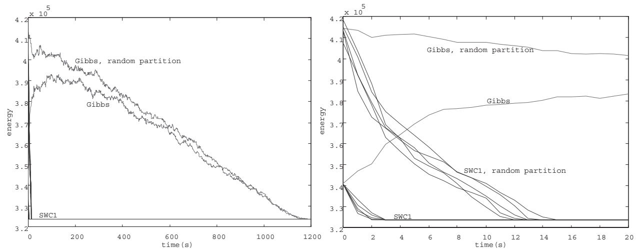


Figure 8. Convergence comparison with Gibbs sampler (upper curves) for the cheetah image. The Gibbs sampler must start with a high temperature and anneal slowly to get to the minimum energy level of our algorithm. Right plot shows a zoom-in view of the first 20 seconds.

In Fig. 8 we plotted the energy vs time (in seconds) of 5 runs of the Swendsen-Wang Cuts (SWC-1) algorithm and one run of the Gibbs sampler on the cheetah image in Fig.2, starting from random partition and 5 runs starting from $\pi =$

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 **IEEE**

$\{G_o\}$ (partition with one subgraph). The Gibbs samplers converges in about 1200 seconds and our algorithm in 15s if starting from random partition, and 3 seconds if starting from a one subgraph partition. The convergence is faster in latter case since the algorithm started from a lower initial energy. Observe that our algorithm converges 400 times faster than the Gibbs sampler.

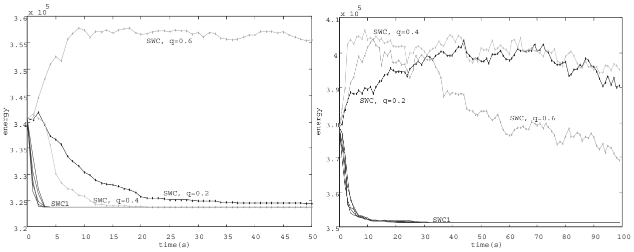


Figure 9. Convergence comparison with SWC without discriminative models ($q_e = 0.2, 0.4, 0.6$) (dotted curves). Cheetah image (left), airplane image (right). The original Swendsen-Wang does not apply in this case.

In Fig.9 we made a comparison of our algorithm with and without the discriminative models, on the cheetah image (left) and airplane image (right), starting from $\pi = \{G_o\}$. For that, we plotted the energy vs time of 5 runs of our SWC-1 algorithm with discriminative models (smooth curves) and without discriminative models (dotted curves), where we fixed the edge weights to constants $q_e = 0.2, 0.4, 0.6$. The convergence slows down significantly, the annealing schedule must be much slower and the initial temperature higher without the discriminative models.

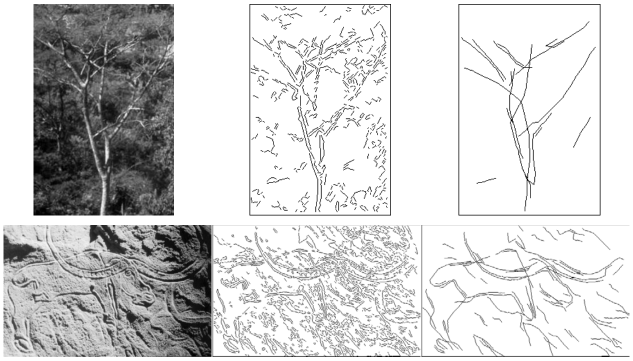


Figure 10. The curve grouping experiment: input image, edge map and grouping result.

In the perceptual grouping experiment we group a map of edgelets obtained from a Canny edge map into long and smooth curves by adding and removing edgelets. The curve prior is based on 3 point histograms learned from hand segmented examples. The likelihood measures the difference in pixels between the input edge map and the grouping result. The graph nodes are the edgelets, and the discriminative probability q_e is based on the 3-point histogram and on the gap that has to be filled between the edgelets. The results are shown in Fig.10.

References

[1] P. J. Green, "Reversible jump MCMC comput. and Bayes. model determination",*Biometrika*,**82**, 711-32 1995

[2] T. Hofmann, J.M. Buhmann, "Pairwise data clustering by deterministic annealing", *PAMI*, 19(1), 1-14, 1997.

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

[4] S. Geman, D. Geman, "Stochastic relaxation, Gibbs distrib., Bayesian restoration of img", *PAMI*, vol. 6, 721-741, 1984.

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

[6] V. Kolmogorov, R. Zabih, "What energy functions can be min. via graph cuts?",*ECCV*, 2002

[7] N. Metropolis, et al. "Eqns of the state calculation by fast computing machines", *J. Chem. Phys*, 21, 1087-91, 1953.

[8] S. Roy, I. Cox, "A max-flow formulation of the n-camera stereo correspondence problem", *ICCV*, 1998.

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

[10] Z.W. Tu, S. C. Zhu, "Image segmentation by data-driven MCMC", *PAMI*, **24**, no. 5, 2002.

[11] R.H. Swendsen, J.S. Wang, "Nonuniversal critical dynamics in MC simulations", *Phys. Rev. Lett.*, **58** no. 2, pp.86-88, 1987

[12] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering *PAMI*, vol.15, 1101-1113, 1993.