Title
Stochastic Graph Partition: Generalizing the Swendsen-Wang Method

Authors
Barbu, Adrian
Zhu, Song-Chun

Publication Date
2003

Stochastic Graph Partition: Generalizing the Swendsen-Wang Method

Adrian Barbu and Song-Chun Zhu

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

Abstract

Vision tasks, such as segmentation, grouping, recognition, and learning, have a "what-goes-with-what" component. It can be formulated as partitioning an adjacency graph into a number of subgraphs, each being a "coherent" visual pattern in the sense of optimizing a Bayesian posterior probability or minimizing an energy functional. In this paper, we generalize Swendsen-Wang (1987)– a well celebrated algorithm in statistical mechanics– for general graph partition. Our objective is to design reversible Markov chain moves in the space of all possible partitions to search for global optimum in the Bayesian framework. We start with an adjacency graph whose vertices are image elements, such as pixels, edgels, small regions, or image bases. For each edge in the graph, we compute a local discriminative probability or probability ratio for how likely the two vertices belong to an underlying visual pattern. These edge probabilities are computed in a bottom-up fashion through previous supervised learning techniques. By turning on/off the edges independently according to these edge probabilities, we obtain a partition of the graph into a number of connected subgraphs. This procedure is in fact a sample from the space of graph partitions. We use it as a proposal (hypothesis) in a probabilistic manner. Thus the algorithm picks up a connected subgraph and flips the label of all its vertices in a single reversible Markov chain jump. In comparison to the classic Gibbs sampler which flips a single vertex at a time, the proposed method achieves: 1). Fast mixing rate – it can flip a large subgraph at a time and the acceptance probability can be made to be one. 2). Short burn-in period – it can walk at low temperature and does not need a long simulated annealing procedure. Thus it is shown to be nearly 100 times faster than the Gibbs sampler and thus produce results in about 1 minute on a PC for image segmentation and curve grouping experiments. The algorithm is tested in image segmentation and curve grouping task, and it is general for many problems in vision and beyond.

Keywords: graph partition, image segmentation, perceptual organization, Swendsen-Wang method, clustering, data-driven Markov chain Monte Carlo.

1 Introduction

Computer vision problems, such as image segmentation, perceptual organization, object recognition, and learning, have a "what goes with what" component. It is a sub-task that groups image elements, such as pixels, edgelets, image bases and textons, into visual patterns, such as regions, curves, objects, and data clusters respectively, so that some grouping criterion is optimized. The problem can be represented in an adjacency graph with the vertices being image elements and edges being spatial relationships. Thus it becomes a graph partition or graph coloring problem. In the literature, graph partition methods are divided in two categories according to the ways in which the optimization criteria are formulated. One is discriminative and the other is called generative.

In a discriminative approach, one computes a similarity (or distance) measure for a pair of adjacent vertices based on their features such as position, orientation, color, and texture etc. This measure specifies how likely the two elements belong to an underlying visual pattern, and is often treated as a weight of the graph edge. The task is to partition the graph into a small number of "coherent" clusters or connected sub-graphs. A widely used discriminative criterion for coherence is compactness:

"*intra-cluster distances are relatively smaller than the inter-cluster distances.*"

Both deterministic, stochastic, and graph theoretical approaches are studied for clustering and partition with various choices of features, similarity measures, and criteria[9, 7, 21, 12, 31]. The discriminative methods are usually convenient to implement and computationally attractive. But they have two serious representational problems which limit their general applicability and robustness.

1. There is no single generally applicable criterion for clustering and graph partition[12]. Natural images contain a mixture of very diverse visual patterns which are "coherent" in different ways. For example, a criterion that prefers compact regions will break elongated curve patterns, and vice versa. Thus we need a set of diverse and competing criteria and models for different patterns.

2. The discriminative methods measure pairwise similarity and it is extremely hard, if not impossible, to capture global properties, for example, global shading effects, perspective projection effects, contour closure etc.

In contrast, generative methods are formulated in a Bayesian framework, and can incorporate a diverse set of models and global prior knowledge. A subgraph is said to be a coherent pattern in the sense that

"*all vertices in a subgraph fit to a chosen family of probability models.*"

Each family of models explains how the 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[25] and three types of curve models are used in perceptual grouping[26].

To achieve globally optimal solutions, generative methods with multiple models have to simulate or maximize Bayesian posterior probabilities using Markov chain Monte Carlo techniques, and thus are computationally intensive. The problem becomes severe if an annealing procedure[17] is used. The main computational bottlenecks are those reversible jumps[4] for the following two types of Markov chain moves[25].

1. Moves type I: the Markov chain should realize reversible model (coherence) selection and switching for each subgraph.

2. Moves type II: the Markov chain must be ergodic in the space of all possible graph partitions. The moves such as split-merge, grouping-ungrouping, and death-birth are made to be reversible and well balanced.

The reversible jumps are implemented by Metropolis-Hastings method[19, 13] and are bridges that connect subspaces (or sub-manifolds) of varying dimensions in the solution (search) space. In this paper we focus on the graph partition (or coloring) moves – type II.

The literature of graph partition and coloring dates back to relaxation-labeling (see [8] and ref therein) – a greedy algorithm that flips vertex label for local consistency. The Gibbs sampler (or "heat bath" in physics) is a stochastic version of relaxation-labeling which can achieve global optima if a simulated annealing procedure[17] is adopted.

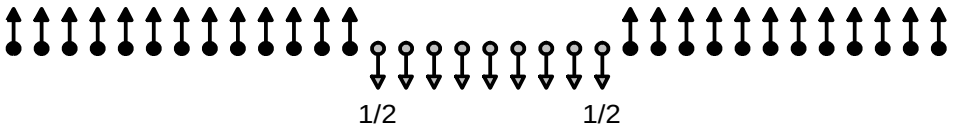


Figure 1: Difficulty in sampling the Ising and Potts models.

The difficulty of sampling the partition space is well reflected in a simple Ising and Potts models[15, 20], which are sometimes used in vision as prior models to enforce region compactness. Figure 1 shows a string of spins whose label $\mathbf{I}$ can be $+1$ (up) and -1 (down). The Ising/Potts model is

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

$\mathbf{1}()$ is an indicator function. $\mathbf{1}(\mathbf{I}_s = \mathbf{I}_t) = 1$ if $\mathbf{I}_s = \mathbf{I}_t$ for two adjacent spins 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 the Gibbs sampler (or Metropolis) flips the -1 spins at the two "cracks". The probability for flipping each spin from -1 to $+1$ is $p_o = 1/2$. Thus to flip a string of n spins ($n = 9$ in Figure 1) from -1 to $+1$ successfully, the expected number of steps is

$$\text{expected steps} = \frac{1}{(1/p_o)^n} = 2^n.$$

This is exponential waiting and is typical in general graph partition and coloring !

A major speedup is achieved by a well celebrated Swendsen-Wang (1987) algorithm[27] in physics. The SW method forms a number of randomly connected subgraphs (i.e. a partition of lattice) by connecting, with a probability $p = 1 - e^{-\beta}$, each pair of adjacent spins of the same label. Then it flips the label of a connected subgraph in a single step. The acceptance probability for such big move is computed to be 1 (see later section for details). For example, all -1 spins in Figure 1 can be flipped to $+1$ in one or a few steps when β is high (low temperature). Thus the SW algorithm achieves fast mixing even at critical temperature and thus does not need a long annealing procedure. [1] Unfortunately, SW is limited to Ising/Potts models and it slows down in the presence of external field (data) as it does not make use of the image (data) information in forming the connected subgraphs.

In this paper, we present a stochastic graph partition algorithm which generalizes SW to general posterior probabilities in vision tasks, such as segmentation and grouping. Our method combines the representational advantages of the generative method and computational efficiency of discriminative models. It is shown to be about 100 times faster than the single site Gibbs sampler, and thus the Markov chain can segment an image in the speed of about 60 seconds.

The basic ideas and contributions of our method are summarized in the following.

1. Given an adjacency graph, we compute a local probability at each edge for how likely the two vertices belong to the same underlying pattern. This is borrowed from the discriminative methods. Then given a current partition which has a number of subgraphs each being a coherent pattern. For each subgraph (pattern), we turn on and off edges inside the subgraph at random according to their associated probabilities, thus each subgraph is broken into a number of randomly connected components. Each component is connected by edges that are turned on. Intuitively, vertices within a component have strong ties and the weak connections (cracks) are broken. This is done in a probabilistic way for reversibility. These connected components are good candidates for re-grouping and re-labeling.

[1] In some worst case when the adjacency graph is a complete graph, the SW method can slow down drastically, but we argue that this is not going to happen in vision as adjacency graphs in segmentation and grouping are always very sparse.

2. Then we re-organize the components by flipping the label of each component at a time. This is like SW, and the move observes the detailed balance equations in general settings. For certain choice, the moves are always accepted with probability 1. Thus our method also be considered a generalized Gibbs sampler.

3. The algorithm "mixes" very fast even at low temperature and thus does not need a long simulated annealing procedure. Therefore we can start from good initial conditions to achieve very short "burn-in" period. In previous work[25], good initialization usually cannot be utilized as high temperature at the early stage brings the Markov chain to random states.

Our method for graph partition is distinct from the various graph cut algorithms in the vision literature.

First, it is distinct from the discriminative methods, such as graph cut and its numerous variations[24, 31], though the ideas of adjacency graph are used as computational heuristics. Our method incorporates many families of image models and global prior knowledge in a generative model setting. The graph components proposed by the discriminative models are coordinated by the Bayesian posterior probability.

Secondly, our method is very different from other recent graph theoretic algorithms for energy minimization[22, 14, 18]. These graph cut algorithms use the maximum flow (or minimum cut) algorithm to find global optima for a class of energy functions in polynomial time. But one has to construct a highly specific graph for a given energy function so that a minimum cut on the graph minimizes the energy. It is shown in [18] that only very limited classes of energy functions are graph representable and thus solvable by such method. In contrast, our algorithm can be applied to optimizing general forms of posterior probabilities.

Our method is an addition to the recently proposed data-driven Markov chain Monte Carlo (DDMCMC) paradigm[25, 26]. The DDMCMC algorithm takes the discriminative methods, such as color and texture clustering, as computational heuristics, and expresses the clustering results in the form of non-parametric probabilities in various spaces of image models. Then these probabilities are used as importance proposal probabilities to guide the reversible jumps, such as model selection, switching and fitting etc (moves type I in our discussion above). In this paper, we add the graph clustering by the discriminative method to expedite moves type II in the graph partition space. These moves are supplemented by other small moves, such as model fitting and boundary diffusion etc in a continuous representation, which are often much easier to compute.

The paper is organized in the following way: We start with a Bayesian (generative) formulation in Section (2). Then we introduce the SW algorithm in Section (3) to set the background. Section (4) presents the discriminative method for sampling the partition

space. Section (5) integrates the discriminative method with Markov chain sampling and proves the ergodicity and detailed balance. Then we show two groups of experiments in Section (6): image segmentation and curve grouping organization. We discuss the computational speed issues. Finally Section (7) concludes the paper with discussions.

2 Bayesian formulation of graph partition

2.1 Partition of graphs

Set partition. Suppose we are given a set of image elements $V = \{v_1, v_2, ..., v_N\}$ such as pixels, edgelets, and textons. The objective is to divide V into an unknown number of n disjoint subsets,

$$V = \cup_{k=1}^{n} V_k, \quad V_k \neq \emptyset, \ V_i \cap V_j = \emptyset \text{ for } i \neq j.$$

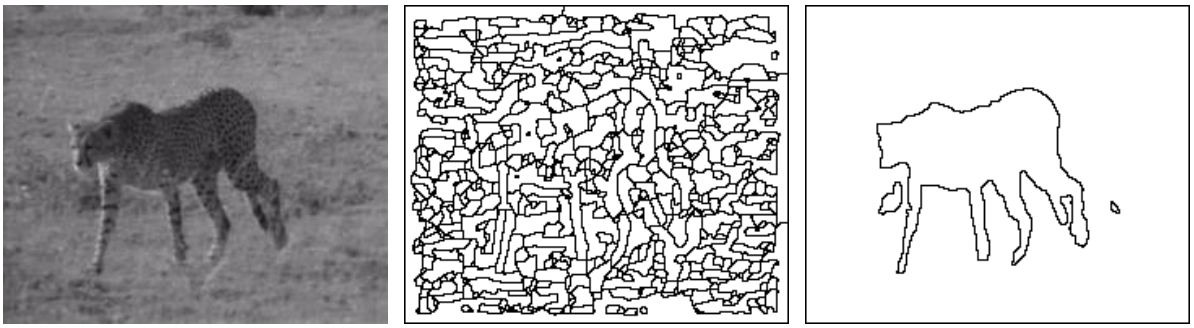
Each subset $V_k, k = 1, 2..., n$ forms a coherent visual pattern in the sense that they fit to a generative probability. We denote a partition with n subsets as π_n,

$$\pi_n = \{V_1, V_2, ..., V_n\} \in \Omega_{\pi_n}, \ n = 1, 2, ..., N.$$

The space of all possible partitions is denoted by

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

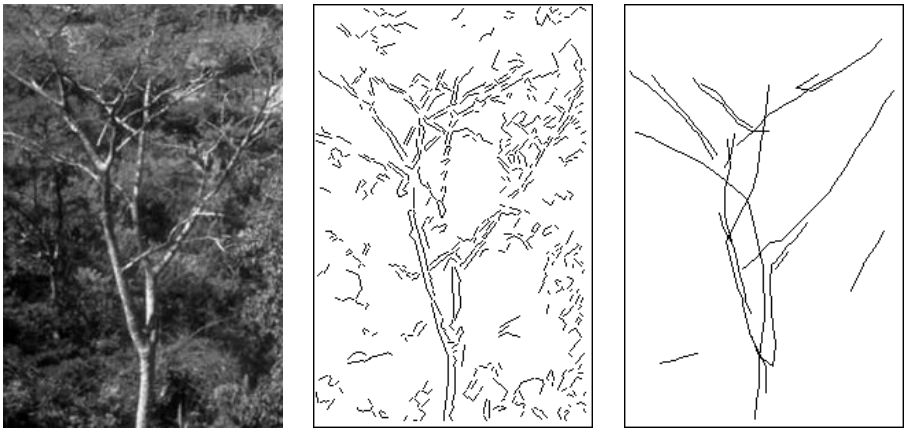
This partition task is common to many vision problems. It is called (1). image segmentation if V_k forms a homogeneous color or texture region, (2). perceptual organization if V_k is a smooth curve, and (3) object recognition if V_k forms an object.



a b c

Figure 2: The image segmentation of a cheetah image. (a). input image (b). a Canny edge detection followed by edge tracing to form small "atomic regions". Each atomic region is treated as a vertex in G_o. (c). a segmentation result.

In this paper, we focus on two cases. The first is segmentation and an example is shown in Figure 2. For an input image in Figure 2.a and we can treat each pixel as v_i. but to reduce the number of elements N we first compute an over-segmentation in Figure 2.b using a Canny edge detection followed by an edge tracing step. In this case the elements $v_i, i = 1, 2, ..., N$ are called "atomic regions" each having nearly constant intensity. The second is curve grouping and an example is shown in Figure 3. The image elements $v_i, i = 1, 2, ..., N$ are the edgelets with positions, orientations and lengths. In general, the elements can also be image wavelets and textons.



a b c

Figure 3: An example of perceptual grouping. (a). An input image of trees, (b). A map of edgelets by Canny edge detection. (c). A number of curves by grouping the edgelets.

Graph partition. In general, people introduce adjacency graphs $G_o =< V, E_o >$ on the set of image elements V. The edges introduce neighborhood structures and spatial constrains on the partition and thus reduce the partition space. The selection of edges E_o balances computational complexity and robustness. In our paper, the edge set E_o is defined as follows.

- For the over-segmented atomic regions shown in Figure 2.b, $e =< v_i, v_j > \in E_o$ if and only if two atomic regions v_i and v_j share boundaries.
- For the map of edgelets shown in Figure 2.c, $e =< v_i, v_j > \in E_o$ if and only if the distance between two edgelets in position and orientation is smaller than certain safe thresholds. The choice of threshold will balance robustness and graph sparsity. when a curve is occluded by a large object, they may lose connection in G_o.

Thus the problem becomes partitioning graph G_o into subgraphs $G_k =< V_k, E_k >, k = 1, 2, ..., n$. Each subgraph $G_k =< V_k, E_k >$ is a *full subgraph* of G_o, i.e., it keeps all the

edges in G_o that connect two vertices in V_k:

$$E_k = \{e = (u, v) \in E_o \mid u, v \in V_k\}, \quad k = 1, 2, ..., n.$$

Note that a subgraph is usually connected but not always. For example, an object in the background can be separated in several pieces due to occlusion. We still use π_n to denote a graph partition as the edges are defined automatically within each subset V_k. The edges between two sets V_i and V_j are denoted by a *cut*

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

To summarize, for a partition π_n, the edges are divided as

$$E_o = [\cup_{k=1}^{n} E_k] \cup [\cup_{i \neq j} C(V_i, V_j)]$$

2.2 Solution space and Markov chain jump steps

In the Bayesian framework, we use a mixture of M classes of models to interpret various visual patterns, e.g. color, texture, shading, curve etc. These types of models are indexed by c,

$$c \in \{C_1, C_2, ..., C_L\} = \Omega_C.$$

The model space is a union of the M models

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

We denote by $\mathbf{I}_v$ the observed image attributes for element v, such as pixel intensity, edge position and orientation. We denote by $\mathbf{I}_V$ the image representation for the set V. Each class of pattern c is defined by a family of probability models $p(\mathbf{I}_V; c, \theta_c)$ specified by a vector valued parameter $\theta \in \Omega_c$. $p(\mathbf{I}_V; c, \theta_c)$ is either parametric or a non-parametric depending on the length of θ_c and can have different dimensions for different types of models.

The inner representation for the observed image elements is

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

The solution space for W is denoted by

$$\Omega = \cup_{n=1}^{N} \{\Omega_{\pi_n} \times \Omega_\theta^n\}.$$

For a fixed n, it is a product of an n-partition space, and n model spaces. Furthermore we can unfold the n model space, and

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

The factorization of the space corresponds to the necessary solution steps: (1). partition the graph by finding a point in Ω_{π_n}; (2). select an image model for each subgraph in Ω_C; (3). fit the model within each model family $\Omega_{c_i}, i = 1, 2, ..., n$.

If we assume the patterns are mutually independent, then the whole image interpretation is subject to a posterior probability,

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

The specific form for the prior model and image models will be selected in experiments and learned off-line. In general these models can be Markov random field models or global spline models, and are beyond what can be minimized by the graph cut with maximum flow algorithms[18].

Our goal is to design ergodic Markov chains which simulate random walks in the solution space Ω and sample from the posterior $p(W|\mathbf{I}_V)$. Usually $p(W|\mathbf{I}_V)$ is very "cold" and sampling from $p(W|\mathbf{I}_V)$ is all we need.

As we mentioned in Section (1), there are two types of jumps bridging the subspaces of different dimensions in Ω.

- Type I is "what is what" – moves in the model space $\Omega_C^n \times \Omega_{c_1} \times \cdots \times \Omega_{c_n}$. For example, switching of model class c and diffusion of parameters θ_c for each subset $V_k, k = 1, 2, ..., n$.
- Type II is "what goes with what" – moves in the partition space Ω_π. For example, split-and-merge, region competition.

Obviously the two steps are tightly coupled.[2] They are reversible jumps[4] realized by Metropolis-Hastings methods[19, 13].

Consider a pair of reversible moves between two states $W = A$ and $W = B$ which are often points in two subspace of different dimensions. The Markov chain design involves two *proposal probabilities* $q(A \to dB) = q(B|A)dB$ – from A to B and $q(B \to dA) = q(A|B)dB$ – from B to A. The proposed move from A to B is accepted with probability

$$\alpha(A \to B) = \min(1, \frac{q(B \to dA)}{q(A \to dB)} \cdot \frac{p(B|\mathbf{I})dB}{p(A|\mathbf{I})dA}) = \min(1, \frac{q(A|B)}{q(B|A)} \cdot \frac{p(B|\mathbf{I})}{p(A|\mathbf{I})}) \tag{4}$$

The Markov chain transition probability is

$$P(A \to dB) = q(A \to dB)\alpha(A \to B), \quad \text{for } A \neq B.$$

[2] 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.

Then it is easy to check that the detailed balance equation is observed

$$p(A)dA\,P(A \to dB) = p(B)dB\,P(B \to dA).$$

When the Markov chain is ergodic and aperiodic in the solution space Ω, then its states follow the posterior $p(W|\mathbf{I})$ after a burn-in period.

As we can see that the effectiveness of Markov chain depends on the design of the proposal probabilities or its ratios $q(A|B)/q(A|B)$. In the literature, many methods are studied to improve Markov chain convergence, such as simulated tempering, dynamic weighting, nevertheless these designs do not make use of the input data $\mathbf{I}$ and thus the Markov chain is close to exhaustive search. The idea of a recent data-driven Markov chain Monte Carlo (DDMCMC) paradigm [33, 25, 26] is to design the proposal probabilities from images using bottom-up (or discriminative) methods. We denote by $D(\mathbf{I})$ the discriminative models (heuristics) from image $\mathbf{I}$. The acceptance probability becomes

$$\alpha(A \to B) = \min(1, \frac{q(A|B, D(\mathbf{I}))}{q(B|A, D(\mathbf{I}))} \cdot \frac{p(B|\mathbf{I})}{p(A|\mathbf{I})}) \tag{5}$$

The objective is to design proposal probabilities which approximate the posterior $q(A|B, D(\mathbf{I})) \approx p(A|\mathbf{I})$ and $q(B|A, D(\mathbf{I})) \approx p(B|\mathbf{I})$ and can be easily sampled, so that the acceptance rate is close to one.

For the type I moves, some *data clustering* methods[9] are used to compute clusters from image $\mathbf{I}$ in each model space Ω_c, $c \in \Omega_C$. Then the clusters are represented in non-parametric form using Parzen windows to make probabilistic proposals for selecting, switching, and fitting models.

For the type II moves, similarly we need to compute *graph clustering* on the partition space Ω_π in a discriminative (data-driven) manner. These probabilities are used for designing smart Markov chain moves in the partition space for fast convergence and mixing. This is studied in the rest of the paper.

3 Background: Swendsen-Wang for Ising/Potts models

The idea of graph clustering is originated from Swendsen-Wang (1987)[27]– an algorithm in statistical mechanics for sampling the Ising/Potts models. The SW method was originally designed to overcome the difficulty that a Gibbs sampler had in sampling the Ising/Potts models. See Figure 1 and discussions in Section (1).

Consider a Potts model in eqn (1) on a 2D lattice. Figure 4 shows two partition states A and B which differ in the labels of the spins inside a box. The SW algorithm realizes a reversible move between A and B in a single step.

Suppose the current Markov chain state is A, the SW algorithm proceeds in the following way according to one of the SW interpretations[30].

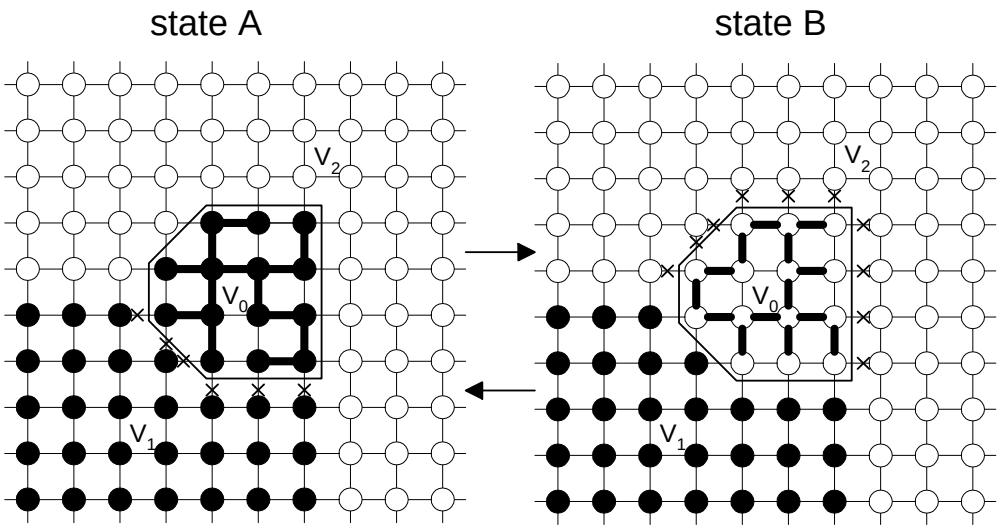


Figure 4: SW algorithm flips a patch of spins in one step for the Ising/Potts models.

1. It selects a vertex s at random, and initializes a set $V_0 = \{s\}$.

2. For any vertex $s \in V_0$, SW finds its neighbor $t \notin V_0$. If s and t have the same label, i.e. $\mathbf{I}_s = \mathbf{I}_t$, then it turns "on" the edge $e =< s, t >$ on with a probability q_o. Otherwise e is turned off. If e is turned on, then it adds t to the set V_0, i.e. $V_0 \leftarrow V_0 \cup \{t\}$. The probability q_o will be decided later.

3. It repeats the above two steps until all edges connecting V_0 to the rest of the graph are turned off. Thus V_0 represents a connected component in Figure 4 (left). The dark edges in V_0 are turned **on**, and other edges are turned **off**.

We denote the remaining black vertices as set V_1, and denote the edges that are turned off between V_o and V_1 as a cut

$$C_{01} = C(V_0, V_1) = \{e =< x, y >: \ \ x \in V_0, \ y \in V_1\}.$$

The cut is illustrated by the crosses in Figure 4.

Obviously there are many ways to arrive at a connected component V_o through the random steps. But they must share a common cut $C(V_0, V_1)$.

Similarly if the Markov chain is currently at state B in Fig. 4 (right), it also has a chance to select a connected component V_o in white. We denote the remaining white vertices as V_2, and the cut between V_0 and V_2 is

$$C_{02} = C(V_0, V_2) = \{e =< x, y >: \ \ x \in V_0, \ y \in V_2\}.$$

So far, we have a pair of states A and B who are different in the labels of V_o. A Metropolis-Hastings method is used to realize a reversible move between them. Though

it is difficult to compute the proposal probabilities $q(A \to B)$ and $q(B \to A)$, one can compute their ratio easily through cancellation.

$$\frac{q(A \to B)}{q(B \to A)} = \frac{(1-q_o)^{|C_{01}|}}{(1-q_o)^{|C_{02}|}} = (1-q_o)^{|C_{01}|-|C_{02}|}. \tag{6}$$

In other words, the probabilities for selecting V_0 in states A and B are the same, except that the cuts are different. Remarkably the probability ratio for $p(A)/p(B)$ is also decided by the cuts through cancellation.

$$\frac{p(A)}{p(B)} = \frac{e^{-\beta|C_{02}|}}{e^{-\beta|C_{01}|}} = e^{\beta(|C_{01}|-|C_{02}|)} \tag{7}$$

The acceptance probability for the move from A to B is,

$$\alpha(A \to B) = \min(1, \frac{q(B \to A)}{q(A \to B)} \cdot \frac{p(B)}{p(A)}) = (\frac{e^{-\beta}}{1-q_o})^{|C_{01}|-|C_{02}|}. \tag{8}$$

By a smart choice of the edge probability

$$q_o = 1 - e^{-\beta},$$

then the proposal from A to B is always accepted with

$$\alpha(A \to B) = 1.$$

As $\beta \propto \frac{1}{T}$ is proportional to the inverse temperature, thus $q_o \to 1$ at low temperature and SW flips a large patch at a time. So SW algorithm can mix very fast at even critical temperature. Although some analysis shows that SW mixed slowly at a worst case when G_o is a complete graph, such case never happens in vision tasks such as image segmentation and curve grouping where the adjacency graphs are very sparsely connected.

In fact, there are many ways to interpret the SW algorithm. Edwards and Sokal (1988) interpreted SW algorithm from the perspective of auxiliary variables and slice sampling[3], and this leads to the idea of partial decoupling in Higdon (1996)[6]. It was applied to image analysis in Barker et al (1998)[2]. Our method bears similarity in spirit to the partial decoupling idea but is different in formulation and is derived in a different way. We should discuss the difference in the discussion section.

Despite the efficiency of SW algorithm, it is not directly applicable to vision tasks for the following reasons:

1. It is limited to Ising and Potts models, while posterior probabilities in vision tasks are of much more complex forms.

2. It is found to be inefficient in the present of external fields (data), as it does not utilize data in the designing the probability q_o for selecting the connected component.

3. It assumes the number of labels n is fixed. The Markov chain does not create new labels in cases where n is unknown.

These limitations will be overcome by our method.

4 Sampling the partition space with discriminative models

In this section, we extend the SW algorithm by incorporating the data information in selecting connected components in general adjacency graph. In the next section, we study the detail balance of the Markov chain design.

Suppose we are given an adjacency graph $G_o =< V, E_o >$. Following the discriminative methods, we extract a number of features $F(v) = (F_1(v), F_2(v), ..., F_a(v))$ at each vertex v. Each edge $e =< s, t >\in E_o$ is augmented with a binary random variable $\mu_e \in \{\text{on}, \text{off}\}$ to indicate whether the edge is turned on or off. In contrast to a constant probability for each edge in the SW algorithm, we compute a discriminative model for $q_e = q(\mu_e = \text{on}|F(s), F(t))$ or a probability ratio based on local features $F(s)$ and $F(t)$,

$$\frac{q(\mu_e = \text{on}|F(s), F(t))}{q(\mu_e = \text{off}|F(s), F(t))}, \quad \text{for } e =< s, t >\in E_o.$$

Such probability ratio can be estimated in a supervised learning stage[16, 5]. It was also shown that techniques like Adaboost can combine a number of weak classifiers to approach the true probability ratio as the number of weak classifiers increases[23].

The probabilities on the edges define a joint probability for any subset of edges $E \subset E_o$,

$$q(E) = \prod_{e\in E} q_e \prod_{e\in E_o - E} (1 - q_e). \tag{9}$$

Therefore E defines a sparse graph $G =< V, E >$ which often consists of a number of n disjoint connected components (subgraphs) $g_1, g_2, ..., g_n$.

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

We denote these connected components by

$$CP = \{V_1, V_2, ..., V_n\}, \quad \cup_{j=1}^{n} V_j = V. \tag{10}$$

This CP is also an n-partition $\pi_n = (V_1, V_2, ..., V_n)$

Later, we are only interested in turning on/off edges within a subgraph $G_l =< V_l, E_l >$, as it is in the previous SW example. Thus we obtain a CP for G_l, and denote it by

$$CP_l = \{V_{l1}, V_{l2}, ..., V_{ln_l}\}, \quad \cup_{j=1}^{n_l} V_{lj} = V_l. \tag{11}$$

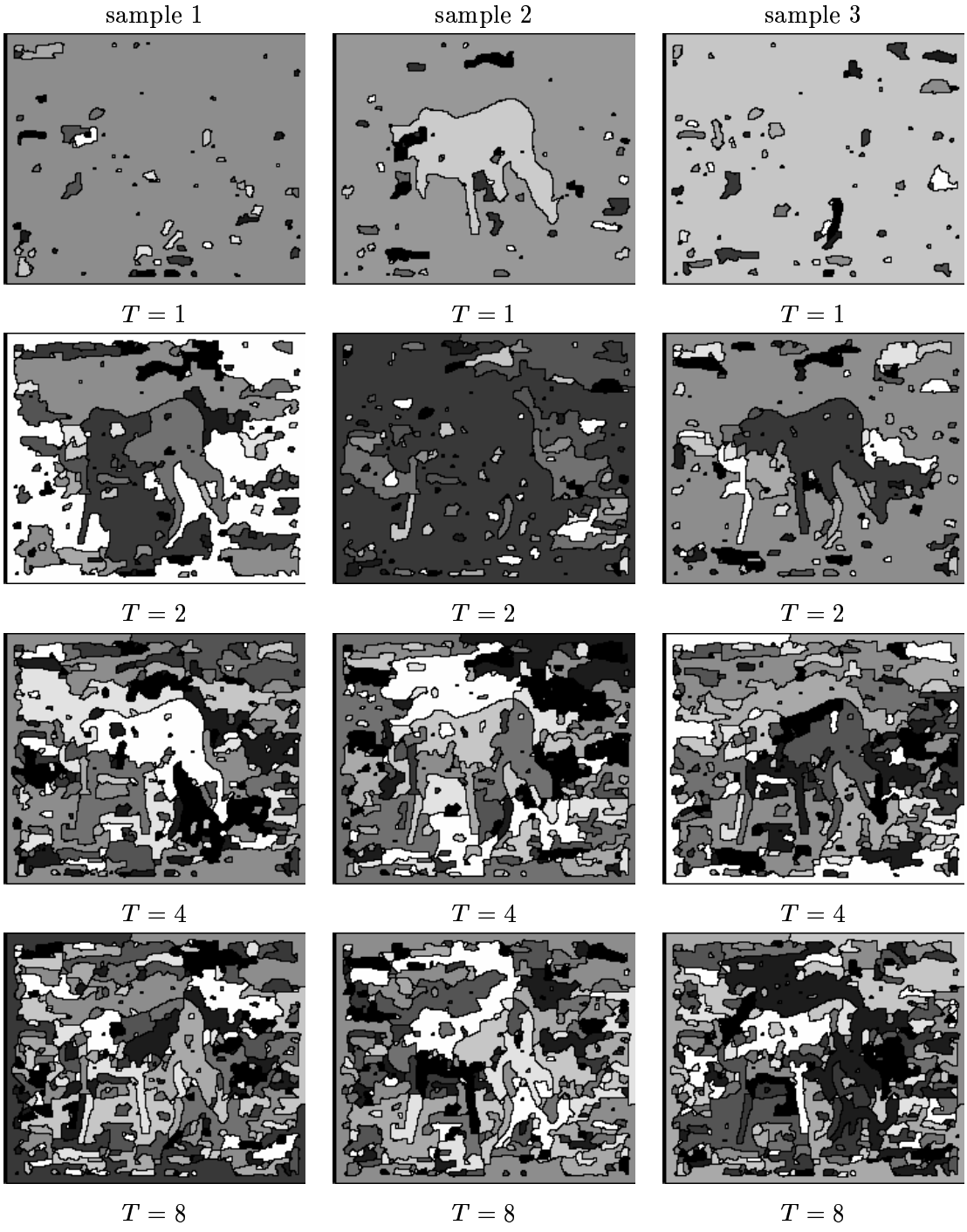


Figure 5: Three samples at temperature $T = 1, 2, 4, 8$ respectively for the discriminative models in the partition space.

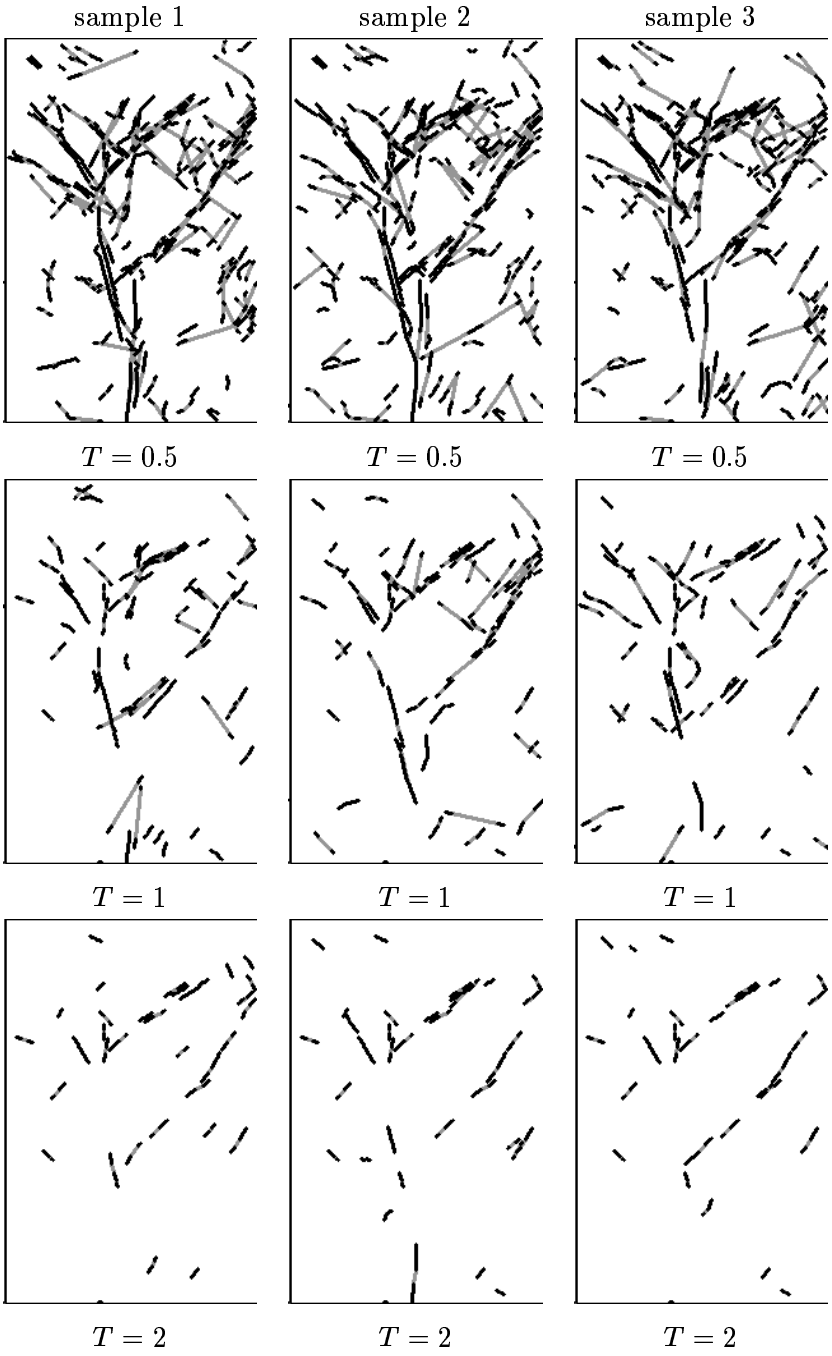


Figure 6: Three samples at temperature $T = 0.5, 1, 2$ respectively for the discriminative models in the partition space.

As the local probabilities are well trained through supervised learning, CP is often a good partition with each connected subgraph corresponding to a pattern. In other words,

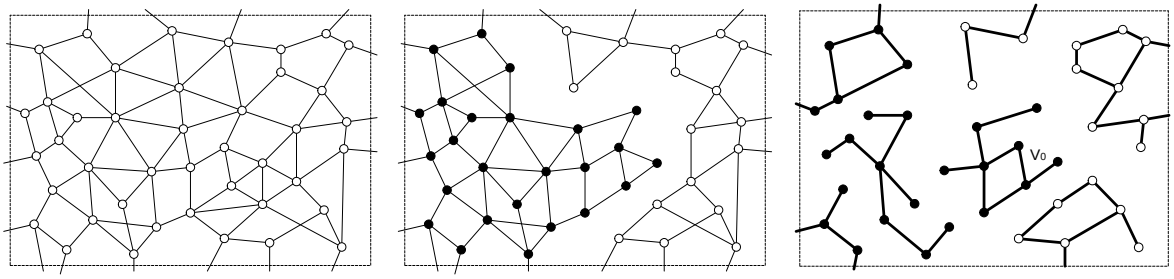
$q(E)$ defines a probability $q(\pi)$ on the partition space Ω_π in a factorized and bottom-up fashion. Obviously the mapping from E to π_n is not one-to-one. Each partition π can be realized by many different edge sets E.

Figure 5 shows some examples of random graph partitions CP for the cheetah image. The adjacency graph is built from the atomic regions in Fig. 2.b. On each row, we show three random partitions CP sampled according to $q(E)$ in equation (9). Each region with the same grey level is a connected component consisting of a number of atomic regions. The edge probability is controlled by a temperature T. When the $T \leq 1$ is low, big regions are formed for the background. When T is high, small regions are formed, as seen in row 4. At a reasonable temperature, various parts of the cheetah are obtained as component candidates for moves.

Similarly, Figure 6 shows three random partitions of curves at three temperatures. The input image and edge maps are shown in Figure 3. The edgels are the graph vertices and each connected component consists of a number of edgels (dark segments) connected by grey lines. We removed the subgraphs with only 1 edgel for clarity. So many edgels are removed at $T = 2$.

As we can see, the discriminative models provide good heuristics for partition, however these partitions are limited by the local features and discriminative models. Global generative models are needed to govern the final partition. In the following section, we show how we may use the CPs as candidates to propose smart moves in the partition space for Markov chain design.

5 Stochastic graph partition by Markov Chain Monte Carlo



a. G_o b. G c. CP

Figure 7: Three typical graphs in our algorithm. a) An adjacency graph G_o as the initial graph, b) A partition of G_o into a number of sub-graphs, this is the Markov chain state G at a time step. c). A discriminative model sample CP obtained by turning on each edge $e \in G$ with probability q_e.

The stochastic graph partition algorithm engages three types of graphs shown in Figure 7. It starts with an adjacency graph $G_o =< V, E_o >$ (Fig.7.a). At each time step it has a partitioned graph G which consists of a number of disjoint *full subgraphs* $G_l =< V_l, E_l >$, $l = 1, 2, ..., n$ with edges between the subgraphs removed and each subgraph is colored differently (Fig.7.b). Then during a move between two partition states, it generates some connected components CP_l by turning on/off the edges in each subgraph G_l. We denote them by $CP = \cup_{l=1}^{n} CP_l$ (see eqn (11). For example the CP in Fig. 7.c has 7 connected components.

We present two versions of the stochastic graph partition algorithm using Figure 8 for illustration.

Stochastic graph partition: SGP-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 π: $G = \cup_{l=1}^{n} G_l$. Denote it state A
2. Repeat,
3. Repeat for each subgraph $G_l =< V_l, E_l >, l = 1, 2, ..., n$ in A
4. For $e \in E_l$, turn $\mu_e =$ on with probability q_e.
5. Partition G_l into n_l connected components: $\{g_{li} =< V_{li}, E_{li} >, i = 1, ..., n_l\}$.
6. Collect all the connected components (*see Fig. 7.c*) in $CP = \{V_{li} : l = 1, ..., n, i = 1, ..., n_l\}$.
7. Select a connected component $V_0 \in CP$ at random with prob $q(V_0 \,|\, CP)$ (*usually a uniform probability* $1/|CP|$) (*Fig.8.a shows an example of* V_0).
8. Propose to reassign V_0 to a subgraph $G_{l'}$, l' follows a probability $q(l'|V_0, A, G_o)$ (*we obtain state B in Fig.8.b if* V_o *is merged to an existing subgraph, or state C in Fig.8.c if* V_0 *is a "stand-alone" new subgraph*).
9. Accept the move with probability $\alpha(A \to B)$ or $\alpha(A \to C)$ in theorem 1.

In the above algorithm, we omit the parallel steps of model switching and fitting for clarity.

The probability $q(l'|V_0, A, G_o), l' = 1, ..., n+1$ depends on V_0, the current state A and the original graph structure G_o. In a trivial design, we may choose

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

We shall discuss a more sophisticated probability $q(l'|V_0, A, G_o)$ shortly so that the

acceptance probability $\alpha(A \to B)$ is always 1 and thus SGP becomes a generalized Gibbs sampler.

The move between states A and B is a typical split-merge operation. It includes the birth and death operations as two special cases.

1. If $l' = n + 1$, V_0 becomes a new pattern. So the move between A and C is a birth operation.

2. Suppose V_0 is a component in subgraph G_l. If $V_0 = V_l$, the whole subgraph G_l is merged into $G_{l'}$. The number of patterns is reduced by one. So it is a death operation.

The second version is different only in the way it selects the set V_0. Instead of sampling all the edges in a current partition, it starts from a single vertex v and grows into a connected component V_0 with a subgraph G_l as we showed for the SW algorithm.

Stochastic graph partition: SGP-2

1. Initialize a graph partition π: $G = \cup_{l=1}^{n} G_l$. Denote it state A.
2. Repeat
3. Select a vertex $v \in V$ at random, e.g. from subgraph G_l. Set $V_0 = \{v\}$
4. Repeat until $\mathcal{C}(V_0, V_l - V_0) \cap \{e, \mu_e = \text{on}\} = \emptyset$
5. Find $e =< s, t > \in \mathcal{C}(V_0, V_l - V_0)$. Let $s \in V_0$,
6. Turn $\mu_e = \text{on}$ with probability q_e, else $\mu_e = \text{off}$
7. If $\mu_e = \text{on}$, then $V_0 \leftarrow V_0 \cup \{t\}$.
8. Propose to merge V_0 to subgraph $G_{l'}$ by sampling from $q(l'|V_0, A, G_o)$. (*we obtain state B in Fig.8.b if V_0 is merged to an existing subgraph, or state C in Fig.8.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.

At each step, both SGP-1 and SGP-2 flip a set of vertices V_0. As we showed in the cheetah (Fig.5) and tree (Fig.6) examples, these sets are often meaningful parts of a big visual pattern suggested by the discriminative models.

In what follows, we show that the acceptance probabilities can be computed easily through cancellation, and they can be made to be 1 through a smart choice of $q(l'|V_0, A, G_o)$ so that the proposals are always accepted. These are stated in the two theorems below.

Theorem 1 *In the above notation, consider a candidate component V_0 selected by* SGP-1-2. *If the proposed move to reassign V_0 from G_l to $G_{l'}$ is accepted with probability*

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

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, $V_{l'} - V_0 = \emptyset$. So $\mathcal{C}(V_0, V_{l'} - V_0) = \emptyset$, and $\prod_{e \in \mathcal{C}(V_0, V_{l'} - V_0)}(1 - q_e) = 1$. $\alpha(A \to B)$ becomes $\alpha(A \to C)$.

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

The cuts $\mathcal{C}(V_0, V_{l'} - V_0)$ are often empty or small so the product probabilities are easy to compute.

Proof.

The proof is rather long but the basic ideas are simple. Our objective is to calculate the proposal probabilities $q(A \to B)$ and $q(B \to A)$ for choosing a particular V_0 among all possible combinations in turning on/off the edges in G_o. Although the two probabilities are very complicated, their ratio $q(B \to A)/q(A \to B)$ is extremely simple through miraculous cancellation. Once this ratio is computed, the conclusion follows straight-forward from the Metropolis-Hastings equation (5).

Firstly, we calculate the proposal probability $q(A \to B)$ in SGP-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'.

$$q(A \to B) = q(B|A, D(\mathbf{I})) = q(V_0|A, D(\mathbf{I}))q(l'|V_0, A, G_o), \tag{12}$$

where $D(\mathbf{I})$ denotes the discriminative models on the edges. For clarity, we discuss the exceptional cases later.

Before a move occurs, each subgraph G_l is broken into a number of connected components CP_l by turning off some 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 8.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 (see the thick edges in Figure 8.a)

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

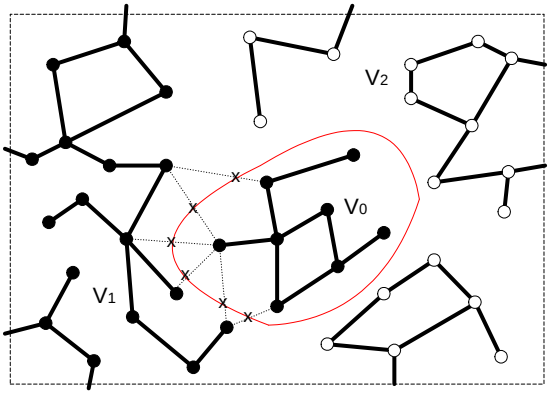
The rest of the edges, which are turned off, are the cuts between a connected component V_{li} and the rest of subgraph, i.e. vertices in $V_l - V_{li}$,

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

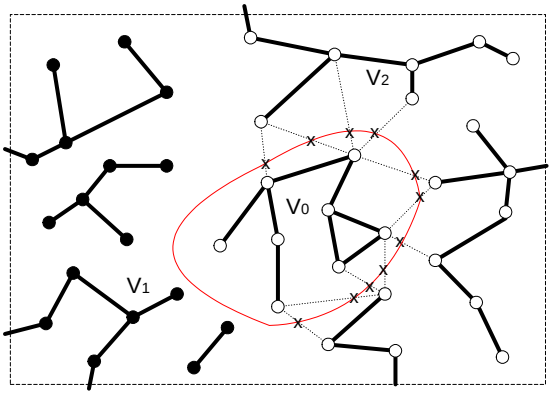
Note that the edges between subgraphs had been turned off before entering state A. The probability for choosing a CP is conditional on state A and $D(\mathbf{I})$,

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

a. A CP of state A



b. A CP of state B



c. A CP of state C

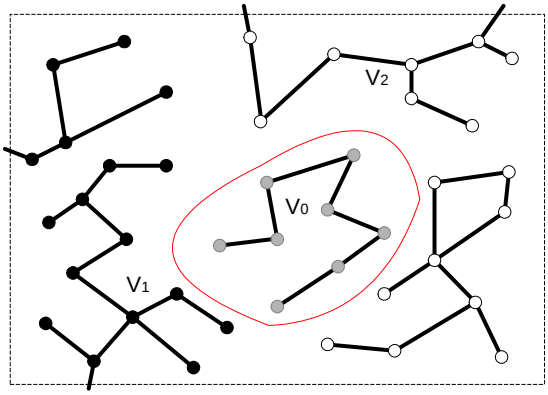


Figure 8: A reversible move between three partition states $\pi = A, B, C$ that are different by a set of vertices V_0. The vertices in the same color belong to a subgraph. The vertices connected by the thick edge form a connected component. A subgraph may have a few connected components.

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

subset where V_0 is among the connected components $V_0 \in CP$, and denote the subset by

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

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

$$\mathcal{C}_{01} = C(V_0, V_1 - V_0).$$

This is illustrated in Figure 8.a by the crosses.

To clarify, all CPs in $\Omega^0_{CP}(A)$ must observe two properties.

1. V_0 must be one connected component in CP. There are usually many ways to make V_0 a connected component.

2. The cut between V_0 and $V_1 - V_0$ must be the same for all CP. That is,

$$C_{01} \subset E_{\text{off}}(A, CP), \quad \forall CP \in \Omega^0_{CP}(A).$$

   These edges must be turned off, otherwise V_0 is connected to other vertices.

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

$$q(V_0|A, D(\mathbf{I})) = \sum_{CP \in \Omega^0_{CP}(A)} q(V_0|CP) q(CP|A, D(\mathbf{I})) \tag{13}$$

$$= \prod_{e \in \mathcal{C}_{01}} (1 - q_e)[\sum_{CP \in \Omega^0_{CP}(A)} q(V_0|CP) \prod_{e \in E_{\text{on}}(A,CP)} q_e \prod_{e \in E_{\text{off}}(A,CP) - \mathcal{C}_{01}} (1 - q_e)]. \tag{14}$$

We can switch the order of the summation and the product $\prod_{e \in C_{01}}(1 - q_e)$ because of property 2 above. We will show that all these terms are canceled out except this product $\prod_{e \in C_{01}}(1 - q_e)$.

Secondly we calculate the proposal probability $q(B \to A)$ in algorithm SGP-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. 8.b). Let $CP(B)$ denote all possible CPs in state B by turning on and off the edges in the n subgraphs at random. Again, we are only interested in those $CP(B)$s that include V_0 as a component,

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

Without loss of generality, we assume that V_0 is a component from the subgraph $G_2 =< V_2, E_2 >$. All the CPs in $\Omega^0_{CP}(B)$ must share a common cut between V_0 and $V_2 - V_0$, denoted by

$$\mathcal{C}_{02} = C(V_0, V_2 - V_0).$$

The cut is illustrated in Figure 8.b by the crosses. Similarly, the probability for selecting V_0 at state B is,

$$q(V_0|B, D(\mathbf{I})) = \sum_{CP \in \Omega^0_{CP}(B)} q(V_0|CP) q(CP|B, D(\mathbf{I})) \tag{15}$$

$$= \prod_{e \in \mathcal{C}_{02}} (1 - q_e)[\sum_{CP \in \Omega^0_{CP}(B)} q(V_0|CP) \prod_{e \in E_{\text{on}}(B,CP)} q_e \prod_{e \in E_{\text{off}}(B,CP) - \mathcal{C}_{02}} (1 - q_e)]. \tag{16}$$

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

$$q(B \to A) = q(A|B, D(\mathbf{I})) = q(V_0|B, D(\mathbf{I})) q(l|V_0, B, G_o). \tag{17}$$

Observation 1. For each $CP \in \Omega^0_{CP}(A)$, then $CP \in \Omega^0_{CP}(B)$ and vice versa. Therefore we have

$$\Omega^0_{CP}(A) = \Omega^0_{CP}(B) \tag{18}$$

That is, for any CP above, the set of edges turned on are the same,

$$E_{\text{on}}(A, CP) = E_{\text{on}}(B, CP) \tag{19}$$

Observation 2. The set of edges turned off are also the same except the cut C_{01} occurs in state A and C_{02} occurs in state B. So

$$E_{\text{off}}(A, CP) - \mathcal{C}_{01} = E_{\text{off}}(B, CP) - \mathcal{C}_{02}. \tag{20}$$

Plug in equations (19) and (20) into equations (14) and (16), we have the probability ratio by cancellation,

$$\frac{q(V_0|B, D(\mathbf{I}))}{q(V_0|A, D(\mathbf{I}))} = \frac{\prod_{e \in \mathcal{C}_{02}} (1 - q_e)}{\prod_{e \in \mathcal{C}_{01}} (1 - q_e)}. \tag{21}$$

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 (5), we obtain $\alpha(A \to B)$ as the theorem states. Thus the move between A and B observe 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.

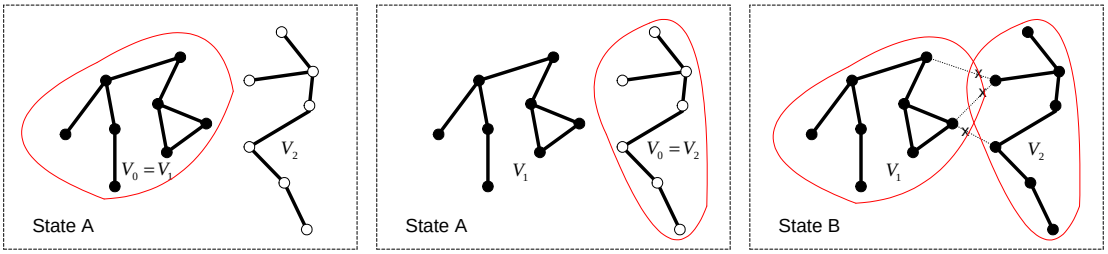


Figure 9: State A has two subgraphs V_1 and V_2 which are merged in state B. There are two paths between A and B. One is to choose $V_0 = V_1$ and the other is to choose $V_0 = V_2$.

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.

- *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.

$$q(A \to B) = q(l' = 2|V_1, A, G_o)q(V_0 = V_1|A, D(\mathbf{I})) + q(l' = 1|V_2, A, G_o)q(V_0 = V_2|A, D(\mathbf{I})) \tag{22}$$

Similarly, we have

$$q(B \to A) = q(l' = 1|V_1, B, G_o)q(V_0 = V_1|B, D(\mathbf{I})) + q(l' = 2|V_2, B, G_o)q(V_0 = V_2|B, D(\mathbf{I})) \tag{23}$$

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,

$$\frac{q(V_0 = V_1|B, D(\mathbf{I}))}{q(V_0 = V_1|A, D(\mathbf{I}))} = \frac{\prod_{e \in \mathcal{C}(V_1, V_2)}(1 - q_e)}{\prod_{e \in \mathcal{C}(V_1, \emptyset)}(1 - q_e)} = \prod_{e \in \mathcal{C}_{12}}(1 - q_e). \tag{24}$$

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

$$\frac{q(V_0 = V_2|B, D(\mathbf{I}))}{q(V_0 = V_2|A, D(\mathbf{I}))} = \frac{\prod_{e \in \mathcal{C}(V_2, V_1)}(1 - q_e)}{\prod_{e \in \mathcal{C}(V_2, \emptyset)}(1 - q_e)} = \prod_{e \in \mathcal{C}_{12}}(1 - q_e). \tag{25}$$

Plug in the above equations, we obtain the ratio,

$$\frac{q(B \to A)}{q(A \to B)} = \prod_{e \in \mathcal{C}_{12}} (1-q_e) \frac{q(l'=1|V_1, B, G_o)q(V_1|A, D(\mathbf{I})) + q(l'=2|V_2, B, G_o)q(V_2|A, D(\mathbf{I}))}{q(l'=2|V_1, A, G_o)q(V_1|A, D(\mathbf{I})) + q(l'=1|V_2, A, G_o)q(V_2|A, D(\mathbf{I}))} \tag{26}$$

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

$$\frac{q(l'=1|V_1, B, G_o)}{q(l'=2|V_1, A, G_o)} = \frac{q(l'=2|V_2, B, G_o)}{q(l'=1|V_2, A, G_o)} \tag{27}$$

This is easily satisfied in general. Then (26) becomes,

$$\frac{q(B \to A)}{q(A \to B)} = \prod_{e \in \mathcal{C}(V_0, V_{l'} - V_0)} (1-q_e) \cdot \frac{q(l'=1|V_1, B, G_o)}{q(l'=2|V_1, A, G_o)} \tag{28}$$

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 of the Markov chain, observe that there is a non-zero probability that any given node is chosen as a connected component V_0. Since this node can then be assigned any other subgraph with non-zero probability, and this is true for all nodes independently, we see that we can get from any partition to any other partition with non-zero probability.

End of Proof.

In practice, the posterior probability ratios $\frac{p(B|\mathbf{I})}{p(A|\mathbf{I}))}$ and $\frac{p(C|\mathbf{I})}{p(A|\mathbf{I}))}$ only involve local computation. For example, in Figure 9 $\frac{p(B|\mathbf{I})}{p(A|\mathbf{I}))}$ only engages model fitting and comparison for $p(V_1 \cup V_2|\mathbf{I})$, versus $p(V_1|\mathbf{I})p(V_2|\mathbf{I})$, as all other regions are not involved in the current move.

In a similar way, one can prove that the same conclusion is true for $SGP-2$. The differences between SGP-1 and SGP-2 are

1. As SGP-2 has a uniform probability to select the initial vertex and thus large vertex set will have a high probability of being selected.

2. SGP-2 reduces the computation slightly as it does not have to sample all edges in the adjacency graph. However, in case when G_o is sparse, the computational improvement is minor.

Now we shall discuss how we choose probability $q(l'|V_0, A, G_o)$ 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 SGP-1 or SGP-2 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 B_{n+1}, V_0 is a new subgraph. In the exceptional case when $V_0 = V_l$, then the state $B_{n+1} = A$ is redundant, so one of them should be eliminated.

We denote the cuts between V_0 and $S_j, j = 1, 2, ..., n+1$ by

$$\mathcal{C}_j = C(V_0, S_j), j = 1, 2, ..., n+1, \quad \text{with } \mathcal{C}(V_0, \emptyset) = \emptyset.$$

$\mathcal{C}_j$ is empty unless S_j is adjacent to V_0 in G_o. Define the weights of the cuts as,

$$\omega_j = \prod_{e \in \mathcal{C}_j} (1 - q_e), \quad \text{and } \omega_j = 1 \text{ if } \mathcal{C}_j = \emptyset.$$

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

$$q(l'|V_0, A, G_o) \propto \omega_{l'} \cdot p(B_{l'} \mid \mathbf{I}). \tag{29}$$

then the proposed move is always accepted with probability one.

Proof. We have

$$q(l'|V_0, A, G_o) = \frac{1}{Z(A)} \cdot \omega_{l'} \cdot p(B_{l'} \mid \mathbf{I}). \tag{30}$$

where $Z(A)$ is a normalization constant, $Z(A) = \sum_{k=1}^{n+1} \omega_k \cdot p(B_k \mid \mathbf{I})$. We get

$$\alpha(A \to B_{l'}) = \min(1, \frac{\omega_{l'}}{\omega_l} \cdot \frac{Z(B_{l'})\omega_l p(A|\mathbf{I})}{Z(A)\omega_{l'} p(B_{l'}|\mathbf{I})} \cdot \frac{p(B_{l'}|\mathbf{I})}{p(A|\mathbf{I})}) = \min(1, \frac{Z(B_{l'})}{Z(A)}) \tag{31}$$

In order to obtain $\alpha(A \to B_{l'}) = 1$ we just need to prove that $Z(A) = Z(B_{l'})$ for $l' = 1, 2, ..., n+1$. In the canonical case, it is trivial to show that $Z(A)$ and $Z(B_{l'})$ are identical being the sum of the same $n+1$ terms.

In the exceptional case, $Z(A)$ and $Z(B_{l'})$ are the same, each having the same n terms, and it is easy to show that this choice of $q(l'|V_0, A, G_o)$ also satisfies condition (27) so that theorem 1 applies.

End of Proof.

Intuitively, we merge V_0 with $S_{l'}$ according to the posterior probability which measures how well they fit to a coherent pattern, modified by a cut factor $\omega_{l'}$ to insure reversibility. In practice, the posteriors $p(B_{l'} \mid \mathbf{I})$ only involve local computation.

Thus we have the third version of the SGP algorithm which is a generalized Gibbs sampler.

Stochastic graph partition: SGP-3

1. Initialize a graph partition $G = \cup_{l=1}^{n} G_l$.
2. Repeat, for a current Markov chain state A.
3. Select a candidate set V_0 as in SGP-1 or SGP-2
4. Draw a random sample l' with probability $q(l'|V_0, A, G_o)$ from (29)
5. Merge V_0 to $S_{l'}$

In contrast to the classic single point Gibbs sampler [11], SGP-3 has the following properties:

1. It flips a large patch of the graph proposed by discriminative models $D(\mathbf{I})$, thus it mixes very rapidly
2. Like SW, it can walk effectively even at low temperature. Thus it does not need a slow annealing procedure and achieves a short burn-in period.

6 Experiments – segmentation and grouping

In this section, we apply SGP-1 to two classical vision problems: image segmentation and curve grouping. We show that the SGP algorithms are about 100 times faster than the Gibbs sampler and the Markov chains converge at about 60 second in segmentation. SGP-2 and SGP-3 have similar performance.

6.1 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 in an image, each being a vertex in G_o.

For an atomic region v_i, we fit its intensity to a Gaussian $p_i = N(\mu_1, \sigma_i^2)$. The discriminative probability q_e for an edge e between two atomic regions v_i and v_j is

$$q_e = 0.1 + 0.9 \exp\{-(KL(p_1||p_2) + KL(p_2||p_1))/2. \quad e =< v_i, v_j > \tag{32}$$

where $KL()$ is the Kullback-Leibler divergence between the probabilities. In general, this q_e can be learned through supervised learning. We adopt three simple image models denoted by $\{C_1, C_2, C_3\}$, and more sophisticated models can be easily added as in [25]. 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 $\mathcal{H}$.

$$\mathbf{J}_1(x, y; \theta) = \mu + \eta, \;\; \eta \sim \mathcal{H}, \;\; \theta_1 = (\mu, \mathcal{H}). \tag{33}$$

The second model C_2 assumes a linear function with additive noise. A linear model:

$$\mathbf{J}_2(x, y; \theta) = \mu + ax + by + \eta, \quad \eta \sim \mathcal{H}, \quad \theta_2 = (\mu, a, b, \mathcal{H}). \tag{34}$$

The third model C_3 assumes a quadratic function with additive noise,

$$\mathbf{J}_3(x, y; \theta) = \mu + ax + by + cx^2 + dxy + ey^2 + \eta, \quad \eta \sim \mathcal{H}, \theta_3 = (\mu, a, b, c, d, e, \mathcal{H}). \tag{35}$$

The selection of model was studied in previous DDMCMC work (Tu and Zhu, 2002). Such models are found to be useful for fitting smoothness regions with global shading effects. The texture is modeled by the non-parametric histogram $\mathcal{H}$. In practice, the discretized $\mathcal{H}$ is represented by a vector $(\mathcal{H}_1, ..., \mathcal{H}_B)$. Let R be a region which is fit by a model c with parameter θ. Let n_j be the number of pixels of R that fall into bin j of the histogram. Then the likelihood probability is

$$P(\mathbf{I}_R;\ c, \theta) \propto \prod_{v \in R} \mathcal{H}(v) = \prod_{j=1}^{B} \mathcal{H}_j^{n_j} \tag{36}$$

a b c

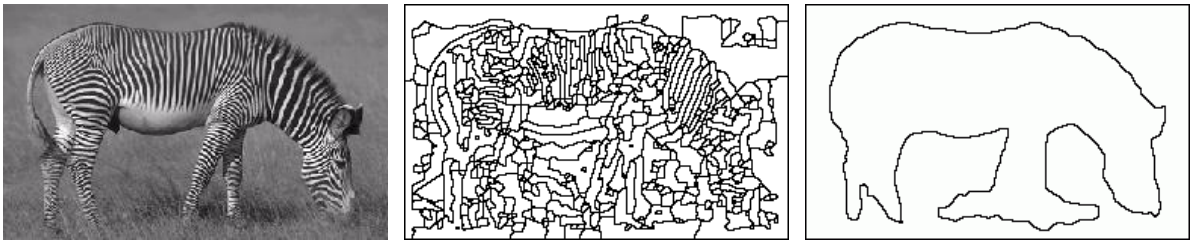
Figure 10: The image segmentation of an artificial image. a. input image b. atomic regions. c. segmentation result.

Like[25], we use the prior $p(W)$ to encourage large and connected regions. Let n be the number of regions, each region may consist of more than one connected sub-regions. That is, several sub-regions may be labeled the same and fit to a shared model. We denote these connected components by $r_1, r_, ..., r_m$, $m \geq n$. The prior is

$$p(W) \propto e^{-\gamma n} e^{-\gamma' m} \prod_{i=1}^{m} e^{-\lambda \text{Area}(r_i)^{0.9}} \tag{37}$$

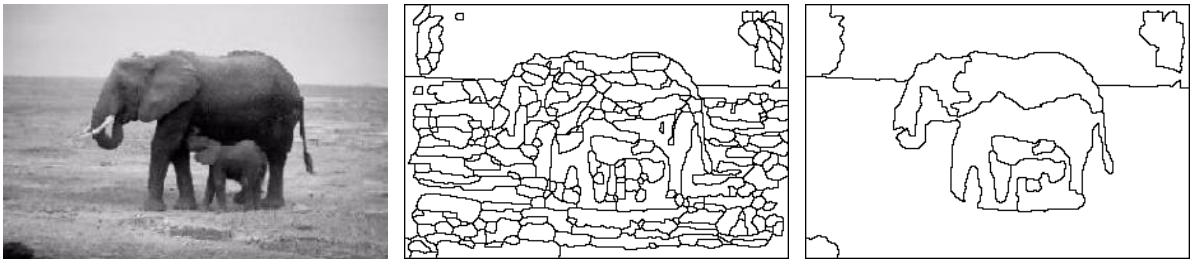
We fix $\gamma = 35, \gamma' = 15$ in our experiments.

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 model



a b c

Figure 11: An example of image segmentation. a. An input zebra image b. atomic regions. c. segmentation result.



a b c

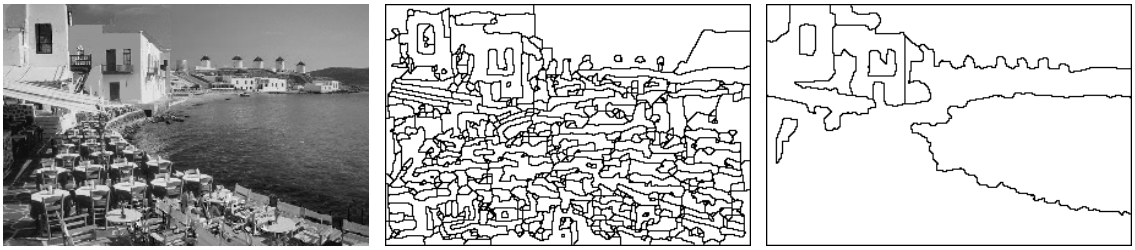
Figure 12: The image segmentation of an elephant image. a. input image b. atomic regions. c. segmentation result.

a b c

Figure 13: The image segmentation of a satellite image. a. input image b. atomic regions. c. segmentation result.

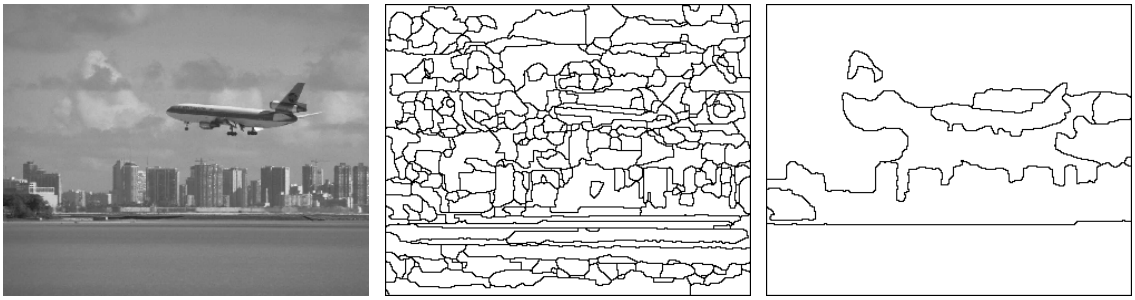
switching, but this is beyond the purpose of our experiments. The image segmentation results obtained from the SGP-1 algorithm are smoothed slightly by a few steps of the region competition equation[32].

We show the five results in Figures 2,10,11, 12,13,14,15. The computational time will be discussed shortly.



a b c

Figure 14: The image segmentation of a coast image. a. input image b. atomic regions. c. segmentation result.



a b c

Figure 15: The image segmentation of an airplane image. a. input image b. atomic regions. c. segmentation result.

6.2 Experiment II: curve grouping

In this experiment we are given an edge map with a number of n edgels. These edgels are obtained using the Canny edge detector followed by fitting long curves by many line segments. Usually we have $n \in [500 - 2000]$ short line segments (edgels of 3-6 pixels long) as vertices V in G_o. We denote them by $v_i = (\mathbf{x}_i^s, \mathbf{x}_i^e), i = 1, 2, ..., N$ with $\mathbf{x}_i^s, \mathbf{x}_i^e$ being the starting and ending points.

Our goal is to group these edgels into an unknown number n subgraphs $V_i, i = 1, 2, ..., n$, each being a chain of edgels. By filling in the gaps between consecutive edgels in V_i we obtain a smooth and continuous curve Γ_i.

Now we choose the likelihood model. In discrete form, the edgel set V in G_o consists of pixels on the edges, denoted by

$$D^{\text{obs}} = \{(i, j) : (i, j) \text{ on } v \in V\}$$

The n continuous curves also contain a set of pixels on curves

$$D = \{(i, j) : (i, j) \text{ on } \Gamma_k, k = 1, 2, ..., n\}$$

We choose the likelihood to be

$$p(D^{\text{obs}}|W) \propto \prod_{(i,j)\in D^{obs}-D} p_0 \prod_{(i,j)\in D-D^{obs}} p_1 = e^{-\lambda_0|D^{obs}-D|-\lambda_1|D-D^{obs}|} \tag{38}$$

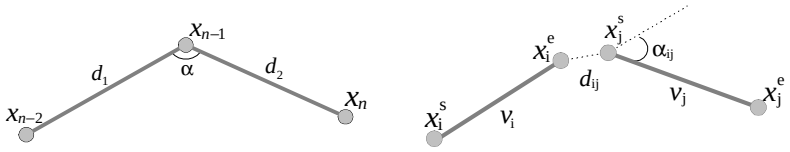
where $p_0 = e^{-\lambda_0} \in [0,1]$ is the probability for detecting a false edge, and penalizes removing too many edges. In contrast, $p_1 = e^{-\lambda_1}$ is the probability for missing an edge and penalizes the gaps in the curves. Each curve is then represented by a list of points $\Gamma_j = (\mathbf{x}_{j1}, \mathbf{x}_{j2}, ..., \mathbf{x}_{jN_j})$. The prior model for a curve group is

$$p(W) \propto \exp\{-\lambda n\} \prod_{i=1}^{n} p(\Gamma_i).$$

Each curve follows a 2nd order Markov chain model.

$$p(\Gamma_i) = p(\mathbf{x}_{i1}, \mathbf{x}_{i2}) \prod_{j=3}^{k} p(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{x}_{j-2}). \tag{39}$$

The probability $p(\mathbf{x}_{i1}, \mathbf{x}_{i2})$ is assumed uniform, while $p(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{x}_{j-2})$ is a two gram represented by a 2-way joint histogram. We compute it by supervised learning from a number of manually parsed images, e.g. from [10].



a b

Figure 16: The joint histograms of $p(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{x}_{j-2})$ contain 6 bins for d_2 and 36 bins for α. There are 6 such histograms, for different values of d_1.

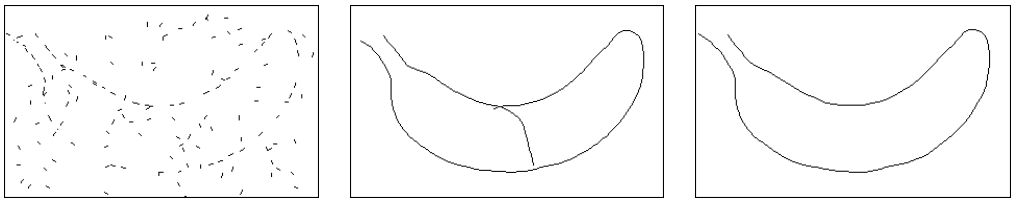
As Figure 16.a shows, we compute three variables: (1). distance $d_1 = |\mathbf{x}_{j-1} - \mathbf{x}_{j-2}|$ (2). distance $d_2 = |\mathbf{x}_j - \mathbf{x}_{j-1}|$, and (3). the angle α. There are 6 histograms, one for the values of d_1 in each of the intervals $[0,2), [2,4), [4,8), [8,16), [16,32), [32,64]$. Each histogram has 6 bins for d_2, in the same range as d_1, and 36 bins for α, each of size 10^o. Thus we have 6 histograms with 6×36 bins each and represent $p(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{x}_{j-2})$ by $p(d_2, \alpha|d_1)$. To avoid empty bins we will start with each bin having one sample in it. There are some details, such as ordering the edgels in a set and computing the relative angle etc. We resolve them in a deterministic way.

To construct G_o, we start with a complete graph on the edgels, and compute an edge strength for any pair $e = (v_i, v_j)$ (see Fig.16.b), based on the gap d_{ij} between the two edgels, and the two gram learned for the prior

$$q_e = 0.99 \cdot p(x_j^s|x_i^e, x_i^s) \cdot p(x_j^e|x_j^s, x_i^e) \cdot e^{-\lambda_1 * d_{ij}} \tag{40}$$

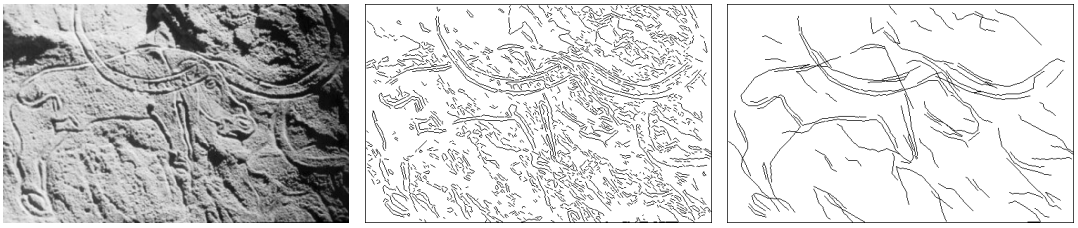
where λ_1 is the gap penalty used in the likelihood equation.

If $q_e < 0.01$ then edge e is removed. We assume this is a very safe threshold to reduce the graph complexity.



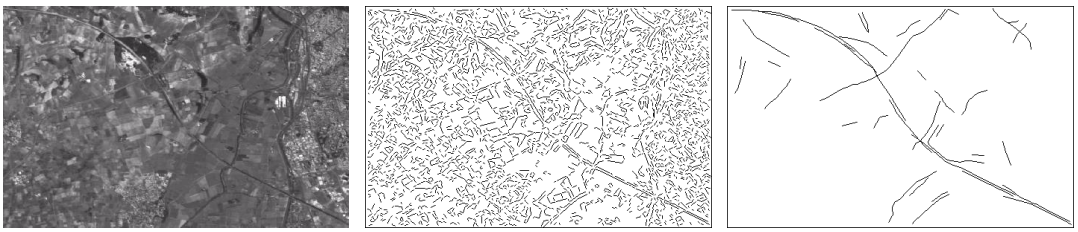
a b c

Figure 17: The curve grouping example: banana. a. input edgel map b. grouping result 1. c. grouping result 2.



a b c

Figure 18: The curve grouping for a stone carving. a. input image, b. edgel map c. grouping result for a number of curves.



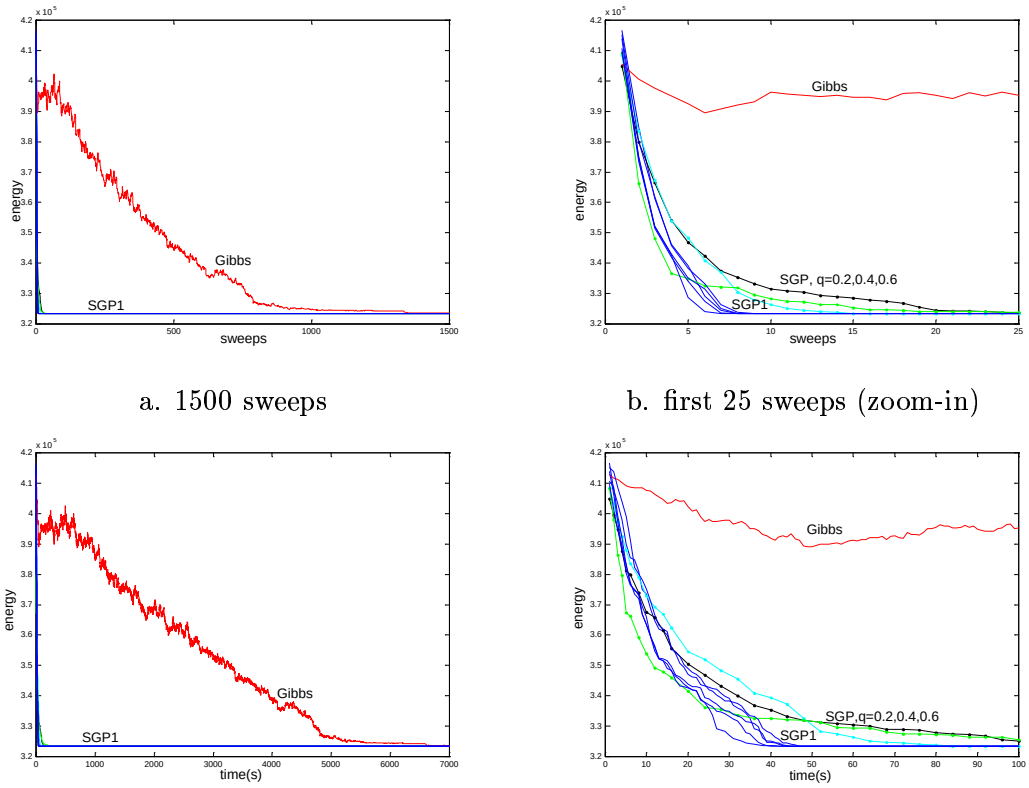
a b c

Figure 19: The curve grouping for a low resolution satellite image. a. input image, b. edgel map c. grouping result for a number of curves for the road.

We display four examples SGP-1 in Fig. 3,17, 18,19. The results are not ideal, mainly because of the simple curve model that we used. In future work, we should introduce more advanced curve models. In fact, most recently we applied the SGP method to grouping parallel curves and trees and more advanced results are in a paper[26].

6.3 Computational speed and comparison

To demonstrate the speedup of the SGP algorithms, we show a speed comparison in Figure 20. We run the SGP-1 algorithm 5 times on the cheetah image in Figure 2 starting with a random partition. We simply assign each atomic region to a subgraph $G_l, l \in \{1, 2, 3, 4, 5\}$.



a. 1500 sweeps

b. first 25 sweeps (zoom-in)

c. 7000 seconds

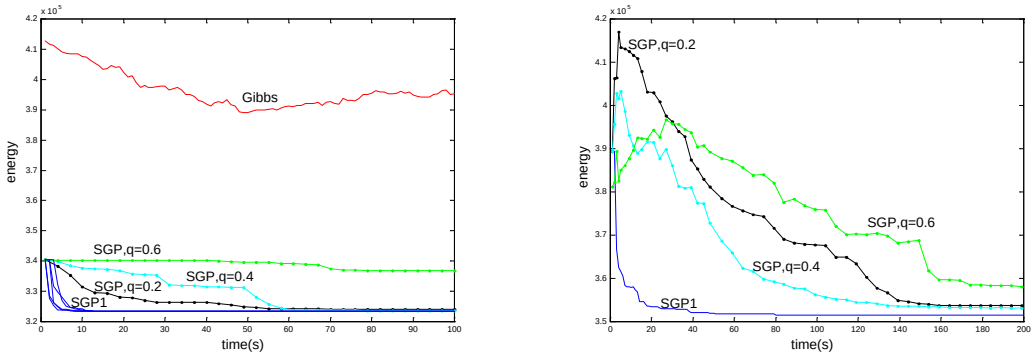
d. first 100 seconds (zoom-in)

Figure 20: Convergence comparison with Gibbs sampler (upper curve). Vertical axis is the energy of the Markov chain state, and the horizontal axis is the number of sweeps for a and b, and running time in seconds for c and d. b is a zoom-in view of the first 25 sweeps and d. is the zoom-in view for the first 100 seconds.

In order to achieve the same low energy level, the Gibbs Sampler has to start with a high temperature $T = 200$ and use an exponential annealing schedule to $T = 0.05$ after 5000 sweeps. Otherwise it can remain stuck at a certain higher energy level. In contrast, the SGP-1 starts at temperature $T = 5$ and decreases to $T = 0.05$ in 20 sweeps. We plot the energy for each run as a function of the number of sweeps in Figure 20.a, and of seconds that elapsed in Figure 20.c.d. As SGP-1 converges much faster, we plot a zoom-in

view of the first 25 sweeps in Figure 20.b, and first 100 seconds in Figure 20.d.

The five SGP-1 runs converge in about $40 - 50$ seconds in a 1.5GHz PC while the Gibbs Sampler converged in 6000 seconds. This is a more than 100 times speed-up!



a. cheetah image b. airplane image

Figure 21: Convergence comparison with SGP1 without discriminative models (edge weights $q_e = 0.2, 0.4, 0.6$, dotted curves). Vertical axis is the energy of the Markov chain state, and the horizontal axis is the running time in seconds.

To study the effect of the discriminative models on convergence, we compare the preformance of our algorithm with and without discriminative models. We made 3 runs of the algorithm without discriminative models, all edges being assigned the same weight, $q_e = 0.2, 0.4, 0.6$ respectively. Observe that the Gibbs sampler is equivalent to SGP with $q_e = 0$. In order to obtain approximately the same final energy, we had to start from a higher temperature and decrease the temperature slowly in these three runs. In Figure 21.a we plotted the energy of the Gibbs sampler starting from random labels, and the energies of 5 SGP1 runs with discriminative models, and the three runs without discriminative models described above (dotted lines), all on the cheetah image in Figure 2. In Figure 21.b we plotted the three runs without discriminative models (dotted lines) and one run of the SGP1, on the airplane image in Figure 15, starting again from a single graph $\pi = \{G_o\}$. We could not compare with the original SW algorithm because it cannot be applied in the general case.

To show the fact that an initial segmentation is useful in obtaining better performance, we started the $SGP1$ runs from a single graph $\pi = \{G_o\}$. The energy of this initial state is much lower than that of a random partition. We see from in Figure 21.a that all 5 runs of the SGP1 algorithm converge to the same energy level in about 15 seconds, compared with 50 seconds when starting from a random partition as in Figure 20.

Compared with the DDMCMC algorithm from [25], our algorithm is about 20-40 times faster. 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 still be 20-40 times faster than the DDMCMC from [25].

We also studied the effect of the discriminative model temperature. We modify the edge weights $\hat{q}_e = q_e^T$ by raising them to a power T. We call it *edge temperature*. Then we ran 5 runs of the SGP-1 algorithm for each of the following temperatures $T = 0.5, 1, 2, 4, 8, 16, 32, 64$ and computed the energy E after 20 sweeps for each run. We compute the average energy $\bar{E}$ over the 5 runs for each T, as a measure of the convergence rate of the algorithm. We plotted the $\log_2 \bar{E}$ as a function of the $\log_2(T)$ above. If T is too small, then too big clusters are being formed and they will be rejected. If T is too big, then too small clusters are being formed and the algorithm is not efficient. This idea was discussed in Figures 5 and 6. Observe though that the best convergence is obtained for $T \in [0.5, 1]$. This is because all the runs started from random partitions, and the grouping happens more often than the ungrouping. If one started with a single-graph partition $\pi = \{G_o\}$, then the best convergence is attained at temperatures $T \in [1, 1.5]$ because now it is more important to split than to merge.

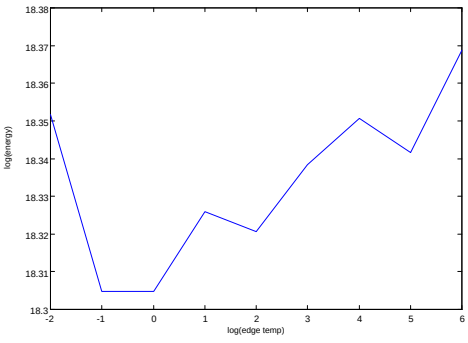


Figure 22: The effects of edge temperature. The average energy level the Markov chains reach after 20 sweeps for a chosen edge temperature, averaged over 5 runs.

7 Discussion

The SW algorithm was very brief (2 pages in Physical Review Letters 1987)[27]. After it was published, a few different views are developed to explain it. One interesting perspective is the auxiliary random variables by (Edward and Sokal, 1988). Each edge $e =< v_i, v_j >$ in G_o is assigned a random variable u_{ij}. Thus one obtains a field of auxiliary variables $U = \{u_{ij} :< v_i, v_j >\}$. Then one augments the posterior to a joint distribution

$p(W, U|\mathbf{I})$. So the Markov chain samples W and U iteratively. This leads to the idea of slice sampling and especially partial decoupling[6]. Barker et al (1998) applied the partial decoupling ideas to image analysis. The SGP algorithms in this paper is related to the partial decoupling idea, but is different in many aspects. First, we derive the SW and SGP from Metropolis-Hastings method with reversible jumps, instead of auxiliary variables. Secondly, we adopt edge probabilities and ratios that should be learned from supervised learning. Thirdly, the SGP algorithm can automatically change the number of partitions n.

The SGP algorithm is an extension to the recent DDMCMC framework[33, 25, 26] by introducing graph clustering in the partition/labeling space. This is combined with the clustering in model space to achieve fast convergence and mixing.

Acknowledgement

The work is supported by an NSF grant IIS-02-44763 and an ONR grant N-00014-02-1-0952. The authors would like to thank Zhuowen Tu and Yingnian Wu for extensive discussions and assistance.

References

[1] A. Amir and M. Lindenbaum, "Ground from figure discrimination", *Computer Vision and Image Understanding*, Vol. 76, No.1, pp.7-18, 1999.

[2] 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.

[3] R.G. Edwards and A.D. Sokal, "Generalization of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm", *Physical Review Letters*, 38, pp 2009-2012, 1988.

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

[5] W.S. Geisler, J.S. Perry, B.J. Super, D.P. Gallogly, "Edge co-occurrence in natural images predicts contour grouping performance", *Vision Research*, 41, pp711-724, 2001.

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

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

[8] R. Hummel and S. Zucker, "On the foundations of relaxation labeling processes", *IEEE Trans. on PAMI*, vol 5, no. 3, pp 267-287, May, 1984.

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

[10] D. Martin, C. Fowlkes, D. Tal, J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics", *Proc. of Int'l Conf. on Comp. Vision*, Vancouver, Canada, July, 2001.

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

[12] Y. Gdalyahu, D. Weinshall, and 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.

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

[14] H. Ishikawa, "Exact optimization for Markov random fields with convex priors", Submitted to *IEEE Trans. on PAMI*, 2001.

[15] E. Ising, "Beitrag zur theorie des ferromagnetismus", *Zeitschrift für Physik*, 31, pp253-258, 1925.

[16] S. Konishi, J.M. Couphlan, A.L. Yuille, and S.C. Zhu, "Fundamental bounds on edge detection: an information theoretic evaluation of different edge cues", *IEEE Trans. on PAMI*, vol.25, no.1, 2003.

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

[18] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?", *Proc. European Conf. on Computer Vision*, pp. 65-81. vol. 3, Copenhagen, Denmark, 2002.

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

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

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

[22] S. Roy and I. Cox, "A maximum-flow formulation of the n-camera stereo correspondence problem", *Proc. Int'l Conf. Computer Vision*, Bombay, India, 1998.

[23] R. E. Schapire, "The boosting approach to machine learning – an overview", MSRI Workshop on Nonlinear Estimation and Classification, 2002.

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

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

[26] 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.

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

[28] J.P. Wang, "Stochastic relaxation on partitions with connected components and its application to image segmentation", *IEEE Trans. PAMI*, **20**, no 6, pp.619-636, 1998.

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

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

[32] 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*, vol. 18, no. 9, pp.884-900, 1996.

[33] S.C. Zhu, R. Zhang, and Z.W. Tu, "Integrating top-down/Bottom-up for object recognition by data-driven Markov chain Monte Carlo", *Proc. of CVPR*, 2000.