

Multigrid and Multi-level Swendsen-Wang Cuts for Hierarchic Graph Partition

Adrian Barbu

University of California, Los Angeles
Computer Science Department
abarbu@ucla.edu

Song-Chun Zhu

University of California, Los Angeles
Statistics Department
sczhu@stat.ucla.edu

Abstract

Many vision tasks can be formulated as partitioning an adjacency graph through optimizing a Bayesian posterior probability p defined on the partition-space. In this paper two approaches are proposed to generalize the Swendsen-Wang cut algorithm[1] for sampling p . The first method is called multigrid SW-cut which runs SW-cut within a sequence of local “attentional” windows and thus simulates conditional probabilities of p in the partition space. The second method is called multi-level SW-cut which projects the adjacency graph into a hierarchical representation with each vertex in the high level graph corresponding to a subgraph at the low level, and runs SW-cut at each level. Thus it simulates conditional probabilities of p at the higher level. Both methods are shown to observe the detailed balance equation and thus provide flexibilities in sampling the posterior probability p . We demonstrate the algorithms in image and motion segmentation with three levels (see Fig.1), and compare the speed improvement of the proposed methods.

1. Introduction: graph partition and coloring

Many vision tasks, such as image segmentation, perceptual grouping, and object recognition can be formulated as graph partitioning (labeling or coloring) problems. A common objective is to partition (or label) an adjacency graph so that the subgraphs represent coherent regions, parts, or objects.

To formulate the problem, we denote by $G = \langle V, E \rangle$ an adjacency graph where $V = \{v_1, v_2, \dots, v_N\}$ is a set of image elements (pixels, edgels, regions, parts, etc.) to be grouped and E is a set of links $e = \langle v_i, v_j \rangle$ representing the spatially adjacency relations between the elements. An n -partition (or coloring) of the elements is a mapping function,

$$\pi_n : V \rightarrow \{1, 2, \dots, n\}^N.$$

We denote it by $\pi_n = \{V_1, V_2, \dots, V_n\}$ where vertices in each subset V_k share the same color and are supposed to form a coherent visual structure depending on the vision

tasks. The *partition (coloring) space* is defined as the set of all possible partitions,

$$\Omega_\pi = \bigcup_{n=1}^{|V|} \Omega_{\pi_n} \quad (1)$$

with Ω_{π_n} being the space of all possible n -partitions.

In vision the earliest algorithm for graph partition is perhaps the relaxation labeling method[8] for solving constraint-satisfaction problems. The well-known Gibbs sampler[7] extends the relaxation labeling method by replacing the hard constraints with “soft” Gibbs energies, and thus samples π from a posterior probability $p(\pi) = p(\pi | \text{Data})$ defined over the entire partition space,

$$\pi \sim p(\pi) \text{ over } \Omega_\pi. \quad (2)$$

Both the relaxation labeling and Gibbs sampling methods flip the label for one vertex at a time, and thus are extremely slow when the elements are strongly coupled.

In the recent vision literature, there were three popular methods for graph partition. One is the normalized cut[14] using graph spectral analysis to optimize approximately a discriminative criterion

Another method is belief propagation[6] which can find a local maximum of the MAP estimate.

The other is the minimum-cut[13, 9] which maps an energy minimization problem to a maximum flow algorithm. Recently the graph cuts algorithm[4] finds minima that are very close to the global minimum for energies with pair and semi-metric potential priors.

Despite their usefulness, these methods can only be applied to limited cases. For example, the maximum flow problem[9] can only be applied when the Gibbs energies satisfy a monotonicity criterion while the graph cuts[4] only works for pair potential priors.

Most recently, a Swendsen-Wang cut method is proposed[1] which generalizes the Swendsen-Wang (1987) method to sampling general posterior probabilities on arbitrary graphs. The SW-cut can flip a coupled subgraph, suggested by bottom-up heuristics, at a single step while observing the detailed balance equations. It is shown[1] to mix rapidly even at low temperature with hundreds of times speedup over the single site Gibbs sampler.

In this paper, we extend the SW-cut in two aspects and thus provide more flexibilities in algorithm design.

1. multi-grid SW-cuts. Given a partition $\pi_n = \{V_1, \dots, V_n\}$, at each step the SW-cut seeks a candidate subset $V_o \in V_i \in \pi_n$ whose label will be flipped. It is costly to select V_o over the entire adjacency graph when G is rather large (e.g. an image lattice has (10^5) vertices). The multi-grid SW-cut first selects a “*attentional window*” of arbitrary shape and size at each step, and then runs the SW-cut within this window. It is shown to sample the conditional probability of p for labeling the vertices within the window given the labels at the rest of the graph. By selecting multi-grid windows (see Fig.4) over time, we observe speed improvements.

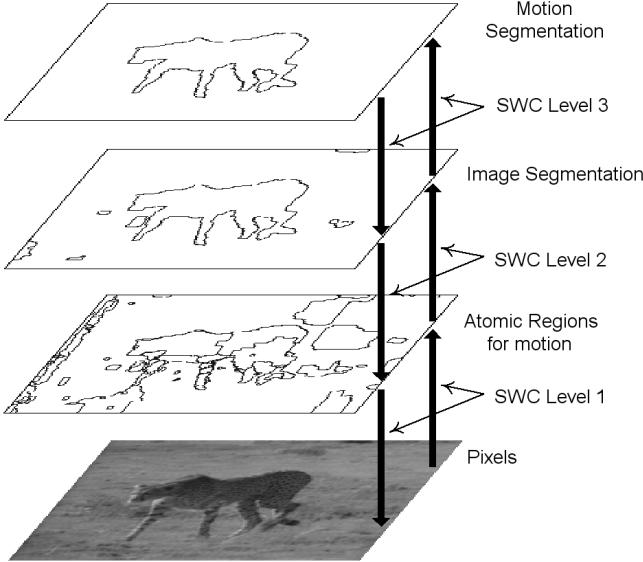


Figure 1. Multi-level SW-cut for image and motion segmentation.

2. Multi-level SW-cuts. Because objects are composed of parts in a hierarchy[2], it is natural to represent them with multiple levels of graphs where each vertex (node) at one level corresponds to a subgraph of smaller elements at the level below. For example, Fig.1 shows an example for partitioning a cheetah image by a hierarchical representation with four levels. The image lattice at level 1 is first partitioned into small regions with nearly constant image and motion and thus form a smaller graph with each region being a node. This graph is further grouped into larger regions with coherent image intensity and coherent motion at the third level. The latter are in turn grouped into a smaller number of moving objects at the fourth level.

The algorithm runs SW-cut at each level, and thus flipping the labels of vertices at one level corresponds to flipping the labels of the corresponding subgraphs (parts) below. This way it samples from the conditional probability

of p given the current subgraphs (parts) on the lower levels. The label change at high levels may correspond to more meaningful moves with larger structures.

The main contribution of the paper is to show that both multigrid and multi-level SW-cuts still observe the detailed balance equations and can be computed easily due to smart cancellation in the probability ratios. The overall algorithm consists of a sequence of multigrid and multi-level SW-cut moves in a cascade, and samples the posterior probability p .

In what follows, we first review the basic ideas of Swendsen-Wang and SW-cut to set the background in Section (2). Then we study the multigrid and multi-level SW-cuts in Section (3). Then we demonstrate the application of the method in a motion segmentation problem in Section (4).

This work was sponsored by NSF SGER grant IIS-0240148.

2. Background: SW and SW-cuts

The original SW-algorithm was proposed in 1987[16] for simulating the Ising and Potts models in statistical mechanics. Unlike the single site Gibbs sampler, SW flips the label of a subgraph at a single step and is known empirically to mix rapidly even near the critical temperature. Recently it has been proven[5] that its mixing time is polynomial in $|V| = N$ for graphs of constant connectivity (i.e. the number of neighbors of each vertex does not grow with size $|V|$).

Unfortunately, the SW method was not very useful for vision tasks for three reasons. Firstly, it is only valid for Ising and Potts models, while posterior probabilities in vision tasks are of much more complex forms. Secondly, it assumes that the number of labels (colors) n is pre-determined, while n is unknown in most vision tasks. Thirdly, it is inefficient in the presence of external fields (data), as it does not utilize data in selecting the subgraph to flip.

These limitations were overcome by the SW-cut method proposed in (Barbu and Zhu, ICCV03)[1]. We briefly introduce the SW-cut and its analysis in the following to set the background.

Let $G = \langle V, E \rangle$ be an arbitrary adjacency graph with $p(\pi)$ being a general probability governing its partition. For each edge $e = \langle s, t \rangle \in E$ in the adjacency graph, we are specified a discriminative probability q_e based on local feature dissimilarity, which measures how likely the two sites s and t belong to different objects, like a boundary detector.

The SW-cut is an iterative algorithm which describes a Markov Chain. Suppose $\pi = \{V_1, \dots, V_n\}$ is the current partition at a certain step, where each subset V_i is the subgraph of nodes with the same color i . The edges between these subgraphs are removed and edges within each subset

are kept. Then for each step the SW-cut proceeds in two phases.

Phase I: For each edge e within a subgraph V_i , i.e. $e = \langle s, t \rangle$ with $s, t \in V_i$, SW-cut turns “off” (removes) at random e with probability q_e . Thus V_i is broken into a number of sub-subsets – the connected components of the remaining graph. They are subgraphs with same label and are connected by the edges remaining “on”. We denote by $\text{ccp}(\pi)$ the set of all connected components from V_1, V_2, \dots, V_n . Obviously there are a combinatorial number of ccp ’s for a given π due to the random procedure.

Then SW-cut randomly selects a component $V_o \in \text{ccp}(\pi)$ (with uniform probability). Suppose $V_o \in V_i$ is selected from V_i , then the edges between V_o and $V_i - V_o$ must have been turned off in the procedure. We denote them by

$$\text{Cut}(V_o, V_i) = \{e = (s, t) : s \in V_o, t \in V_i, t \notin V_o\}, \forall i.$$

Fig. 2 illustrates two partitions π_A and π_B of a lattice which differ in the labeling of a set of vertices V_o within the polygon. In π_A , V_o is selected from V_1 , and in π_B , V_o is selected from V_2 . Then $\text{Cut}(V_o, V_1)$ and $\text{Cut}(V_o, V_2)$ are shown by the crosses in two states respectively.

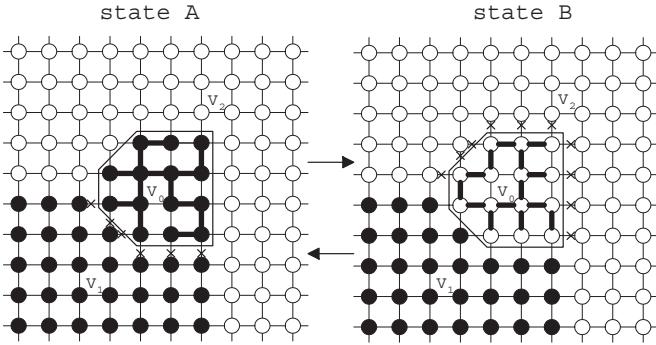


Figure 2. Flipping the labels of a component V_o at a single step between two states π_A and π_B . See text for explanation

Phase II. SW-cut decides the new label of V_o probabilistically (see eqn. (3)). Let $\ell_{\text{new}}(V_o) = j \in \{1, 2, \dots, n+1\}$ be the new label, then we obtain a new partition

$$\pi_j^* = \{V_k : k \neq i, k \neq j\} \cup \{V_i - V_o, V_j \cup V_o\}, \forall j.$$

If $V_o = V_i$, i.e. the whole subgraph V_i is selected by chance, then one label is reduced $|\pi^*| = n-1$, and conversely if $j = n+1$ then one label is introduced $|\pi^*| = n+1$.

The main contributions of the SW-cut[1] are given in the following two observations.

Observation 1. Let π_A and π_B be any two consecutive states in the SW-cut (see Fig.2). We denote by $Q(V_o|\pi_A)$ and $Q(V_o|\pi_B)$ the probabilities that the random procedure selects V_o at states π_A and π_B respectively. The probability ratios can be computed easily due to cancellation.[1]

Theorem 1 *The probability ratio for proposing V_o as the candidate subgraph at two states π_A and π_B is*

$$\frac{Q(V_o|\pi_A)}{Q(V_o|\pi_B)} = \frac{\prod_{e \in \text{Cut}(V_o, V_1)} q_e}{\prod_{e \in \text{Cut}(V_o, V_2)} q_e}.$$

This is not a trivial conclusion because there are combinatorial number of ways for connecting V_o and the rest of the graphs in the random procedure.

Observation 2. Suppose that $\pi_j^*, j = 1, 2, \dots, n+1$ are the possible new states from π , and SW-cut selects the new label j according to its posterior probability $p(\pi_j^*)$ like the Gibbs sampler, but with a rectifying factor $\prod_{e \in \text{Cut}(V_o, V_j)} q_e$, then we have the following conclusion.

Theorem 2 *At state π , if the new label of V_o , i.e. the next partition state π_j^* is selected according to probability*

$$Q(\ell_{\text{new}}(V_o) = j | V_o, \pi) = \frac{1}{C} \prod_{e \in \text{Cut}(V_o, V_j)} q_e \cdot p(\pi_j^*), \forall j$$

then the proposed move will always be accepted. (3)

In the above formula, C is a normalizing constant for the $n+1$ probabilities.

For a special case, the reversible move between π_A and π_B in Fig.2 observes the following theorem.

Theorem 3 *The proposal probability ratio between the two states π_A and π_B is*

$$\begin{aligned} \frac{Q(\pi_A \rightarrow \pi_B)}{Q(\pi_B \rightarrow \pi_A)} &= \frac{Q(V_o|\pi_A)}{Q(V_o|\pi_B)} \cdot \frac{Q(\ell_{\text{new}}(V_o) = 2 | V_o, \pi_A)}{Q(\ell_{\text{new}}(V_o) = 1 | V_o, \pi_B)} \\ &= \frac{p(\pi_B)}{p(\pi_A)}. \end{aligned}$$

So, the acceptance probability in the Metropolis-Hastings[11] step is

$$A(\pi_A \rightarrow \pi_B) = \min(1, \frac{Q(\pi_A \rightarrow \pi_B)}{Q(\pi_B \rightarrow \pi_A)} \cdot \frac{p(\pi_A)}{p(\pi_B)}) = 1.$$

The SW-cut was shown to speed up the Gibbs sampler by $O(10^2)$ folds in image segmentation problems.[1]

3. Multigrid and multi-level SW-cut

The essence of the SW-cut is a Markov chain $\mathcal{MC} = \langle \nu, \mathcal{K}, p \rangle$ which visits a sequence of states in the partition space Ω_π over time t ,

$$\pi(0), \pi(1), \dots, \pi(t) \in \Omega_\pi.$$

The \mathcal{MC} consists of three elements. (1). An initial probability $\nu(\pi)$ with $\pi(0) \sim \nu(\pi)$. (2). A transition kernel $\mathcal{K}(\pi, \pi')$ which is a conditional probability for moving from

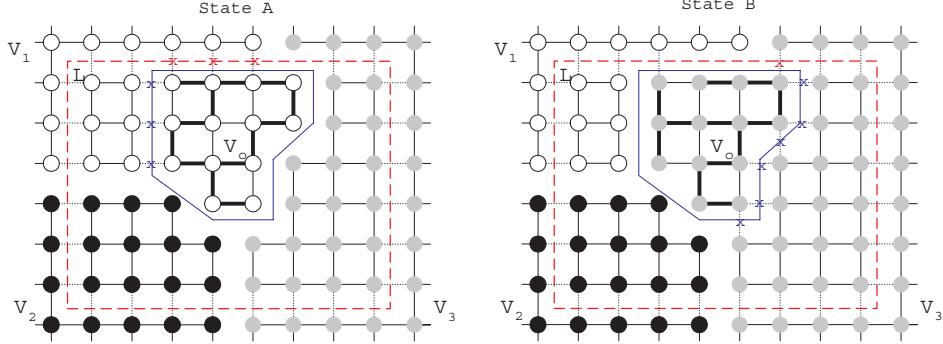


Figure 3. Multigrid SW-cut: run SW-cut within an “attention” window Λ with the rest of the labels fixed, and it realizes a reversible move between two states π_A and π_B by flipping the label of $V_o \subset V_\Lambda$.

state $\pi(t) = \pi$ to $\pi(t+1) = \pi'$. (3). An invariance probability $p(\pi)$ which is the Bayesian posterior probability for the partitions in a vision task.

The three theorems in the previous section ensure that the SW-cut design observes the detailed balance equations

$$p(\pi)\mathcal{K}(\pi, \pi') = p(\pi')\mathcal{K}(\pi', \pi), \quad \forall \pi', \pi. \quad (4)$$

This is a sufficient condition for $p(\pi)$ being the invariant probability of the kernel \mathcal{K} ,

$$\sum p(\pi)\mathcal{K}(\pi, \pi') = p(\pi'). \quad (5)$$

Once it converges, the SW-cut simulates fair samples from $p(\pi)$

The SW-cut is characterized by three selections in its design.

(I). The discriminative proposal probabilities defined on the edges of the adjacency graph $G = \langle V, E \rangle$. $q(\pi) = \prod_{e \in E} q_e$ is a factorized approximation to $p(\pi)$ and it influences the formation connected components CCP, and thus the candidate component V_o .

(II) The uniform probability for selecting V_o from a connected component $V_o \in \text{CCP}$.

(III) The reassignment probability $Q(\ell_{\text{new}}(V_o)|V_o, \pi_A)$ for the new label of the connected component V_o .

We propose to extend the SW-cut for speed and generality by introducing the multigrid and multilevel SW-cuts which provide more flexible means for selecting V_o 's and $q(\pi)$'s.

In summary, the two extensions are new directions for sampling $p(\pi)$

1. The multigrid SW-cuts simulates Markov chain \mathcal{MC}_{mg} with kernel \mathcal{K}_{mg} sampling the conditional probabilities of $p(\pi)$.
2. The multi-level SW-cuts simulates Markov chain \mathcal{MC}_{ml} with kernel \mathcal{K}_{ml} sampling the conditional probabilities of $p(\pi)$ at the higher levels, and the full posterior at the lower level.

Both \mathcal{MC}_{mg} and \mathcal{MC}_{ml} satisfy the detailed balance equations in (4), as it will be shown in the following sections. The proofs are based on the following result.

Let $p(x, y)$ be a two dimensional probability, and \mathcal{K} be a Markov kernel sampling its conditional probability $p(x|y)$ (or $p(y|x)$). Thus it observes the detailed balance equation,

$$p(x|y)\mathcal{K}(x, x') = p(x'|y)\mathcal{K}_1(x', x), \quad \forall x, x'. \quad (6)$$

Theorem 4 *In the above notation, \mathcal{K} observes the general detailed balance equations after augmenting y*

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

Proof. If $y = y'$, then it is straightforward. If $y \neq y'$ then $\mathcal{K}((x, y), (x', y')) = \mathcal{K}((x', y'), (x, y)) = 0$ because there is no way to go from state (x, y) to state (x', y') .

The conclusion of this theorem is that an algorithm which is reversible when sampling from a conditional probability is also reversible for sampling the full probability.

3.1. SW-cuts at multigrid

We first study the multigrid SW-cut. Recall that in each step the SW-cut turns off, probabilistically, the edges in the entire adjacency graph, and this could be less effective especially when G is very large. The concept of multigrid SW-cut is to allow us to select certain “attentional” windows and run the SW-cut within the window. Thus it provides flexibility in designing a “visiting scheme” by selecting windows of various sizes and locations over time. For example, Fig. 4 shows windows in a multigrid arrangement.

Let $G = \langle V, E \rangle$ be the adjacency graph, $\pi = \{V_1, \dots, V_n\}$ the current partition, and Λ an attentional window of arbitrary size and shape. Λ divides the vertices into two subsets $V = V_\Lambda \cup V_{\bar{\Lambda}}$ for vertices inside and outside the

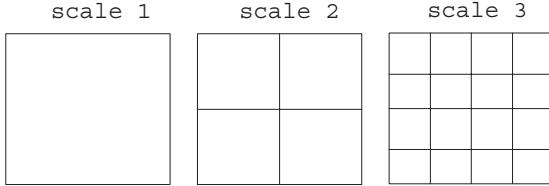


Figure 4. Selecting windows in a multigrid scheme

window respectively. For example, Fig.3 displays a rectangular window Λ (in red) in a lattice G .

The Λ further removes some edges within each subset $V_i, i = 1, 2, \dots, n$, and we denote them by,

$$\text{Cut}(V_i|\Lambda) = \{e = \langle s, t \rangle : s \in V_i \cap V_\Lambda, t \in V_i \cap V_\Lambda\}.$$

For example, in Fig.3 the window Λ intersects with three subsets V_1 (white), V_2 (black), and V_3 (grey), and all edges crossing the (red) rectangle window are removed.

We divide the labeling (coloring or partition) of the vertices V into two parts

$$\pi(V) = (\pi(V_\Lambda), \pi(V_{\bar{\Lambda}})) \quad (7)$$

We fix $\pi(V_{\bar{\Lambda}})$ as boundary condition, and sample the labels of vertices within the window by SW-cut.

To summarize, the multigrid SW-cut iterates the following three steps

1. it selects a window Λ of certain size and shape following a probability $\Lambda \sim q(\Lambda)$.

2. For any edges within each subgraph inside the window,

$$e = \langle s, t \rangle, s, t \in \Lambda, \ell_s = \ell_t,$$

it turns off edge e with probability q_e . Thus it obtains a set of connected components $\text{ccp}(V_\Lambda)$.

3. It selects $V_o \in \text{ccp}(V_\Lambda)$ as a connected component and flips its label according to probability

$$Q(\ell_{\text{new}}(V_o) = j | V_o, \pi) = \frac{1}{C} \prod_{e \in \text{Cut}_j} q_e \cdot p(\pi_j^*), \forall j \quad (8)$$

where π_j^* is the partition by assigning V_o to label j , and $\text{Cut}_j = \text{Cut}(V_o, V_j) - \text{Cut}(V_j|\Lambda)$.

For example, Fig. 3 illustrates a reversible move by flipping a connected component V_o (within the blue polygon) between two states π_A and π_B . Cut_1 and Cut_3 are shown by the blue crosses which are removed by the random procedure.

Following the same procedure as in the previous SW-cut, we can derive the proposal probability ratio for selecting V_o in the two states within Λ .

Theorem 5 *The probability ratio for proposing V_o as the candidate subgraph within window Λ at two states π_A and π_B is*

$$\frac{Q(V_o|\pi_A, \Lambda)}{Q(V_o|\pi_B, \Lambda)} = \frac{\prod_{e \in \text{Cut}(V_o, V_1) - \text{Cut}(V_1|\Lambda)} q_e}{\prod_{e \in \text{Cut}(V_o, V_3) - \text{Cut}(V_3|\Lambda)} q_e}.$$

The difference between this ratio and the ratio in theorem 1 is that some edges (see the red crosses in Fig.3) no longer participate the computation.

Following the probability in eqn.(8) for the new labels, we can prove that it simulates the conditional probability,

$$\pi(V_\Lambda) \sim p(\pi(V_\Lambda)|\pi(V_{\bar{\Lambda}})).$$

Theorem 6 *The multigrid SW-cut within window Λ simulates a Markov kernel*

$$\mathcal{K}(\Lambda) = \mathcal{K}(\pi(V_\Lambda), \pi'(V_\Lambda)|\pi(V_{\bar{\Lambda}})), \quad (9)$$

$$p(\pi(V_\Lambda)|\pi(V_{\bar{\Lambda}}))\mathcal{K}(\pi, \pi') = p(\pi'(V_\Lambda)|\pi(V_{\bar{\Lambda}}))\mathcal{K}(\pi', \pi).$$

Following theorem 4, we have $\mathcal{K}(\Lambda)$ satisfies the general detailed balance equation in eqn.(4).

3.2. SW-cuts at multi-level

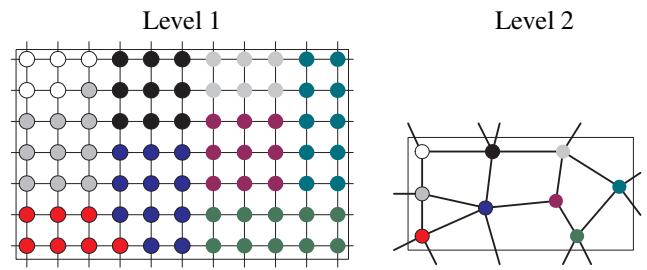


Figure 5. Multi-level SW-cut: run SW-cut at two levels.

Now we add a multi-level SW-cut mechanism. Suppose at state $\pi = \{V_1, V_2, \dots, V_n\}$, we “freeze” some subsets $A_k, k \in \{1, \dots, m\}$ such that for any k , $A_k \subset V_i$ for some i . This way, the vertices in each A_k are locked to have the same label. The subsets A_k can represent an intermediary segmentation. For example, for motion segmentation, it is useful to get an intensity segmentation A and group the intensity regions A_k into coherently moving objects.

Thus $G = G^{(1)}$ is reduced to a smaller adjacency graph $G^{(2)} = \langle U, F \rangle$. U is the set of vertices

$$U = \{u_1, \dots, u_m\}, u_k = A_k, k = 1, 2, \dots, m.$$

F is the adjacency relation between the subsets A_k in G .

$$F = \{f = \langle u_i, u_j \rangle : \text{Cut}(A_i, A_j) \neq \emptyset\}.$$

Fig.5 illustrates an example with $m = 9$. We run the SW-cut on level 2 based on new discriminative heuristics $q^{(2)}$ which

measure the similarity of A_i, A_j , $q^{(2)}(\pi(U)) = \prod_{f \in F} q_f^{(2)}$. In general, these heuristics are more informative than the lower level, so the SW-cuts moves are more meaningful and the convergence is faster.

The partition space for graph $G^{(2)}$ is a projection of Ω_π , $\Omega_\pi(G^{(2)}) = \{\pi : \ell(s) = \ell(t), \forall s, t \in A_i, i = 1, 2, \dots, m\}$

Obviously, the SW-cut on level 2 simulates a Markov chain with kernel $\mathcal{K}^{(2)}$ which has invariant probability $p(\pi(U)|A)$, the probability of $p(\pi)$ conditional on the relations $\ell(s) = \ell(u_i)$ for all $s \in A_i$ and all i .

Following theorem 4, we have that $\mathcal{K}^{(2)}$ satisfies the general detailed balance equation (4).

Summary. Suppose we design a visiting scheme for selecting the windows $\Lambda \sim q_w(\Lambda)$ and level $\sigma \sim q_l(\sigma)$ over time. Then the generalized SW-cut has a mixed Markov kernel

$$\mathcal{K} = \sum_{\sigma} \sum_{\Lambda} q_l(\sigma) q_w(\Lambda) \mathcal{K}^{(\sigma)}(\Lambda).$$

As each $\mathcal{K}^{(\sigma)}(\Lambda)$ observes the detailed balance equations, so is \mathcal{K} . When the windows cover the entire graph, it is also irreducible and its states follows $p(\pi)$ at convergence.

4. Example: image and motion segmentation

We apply the multigrid and multi-level SW-cuts above to a vision problem that integrate image and motion segmentation, and compare the speeds of various visiting schemes.

4.1. Representation and Bayesian formulation

The motion estimation[18, 3] and motion segmentation[12, 17] problems have been intensively studied in the vision literature. With the power of the SW-cut above, we entertain a more general setting for the problem by integrating both image and motion cues in a three level representation (see Fig.1).

At level 3, a scene has $M^{(3)}$ moving objects

$$\{(O_i, \theta_i) : i = 1, \dots, M^{(3)}\}$$

e.g. the cheetah and background, and each object has deformable motion of type rigid plus deformation (Markov Random Field), specified by parameters θ_i , similar in nature to [3]. An object O_i consists of a number of image regions at level 2

$$\{(R_{ij}, H_{ij}, H_{ij}^p) : j = 1, \dots, M_i^{(2)}\}, \forall i.$$

An image region has a coherent intensity model H_{ij} and a model for image intensity prediction by motion H_{ij}^p , both modeled by non-parametric histograms. An image region R_{ij} in turn contains a number of atomic regions with constant motion velocity (dx, dy) at level 1 and constant intensity.

$$\{r_{ijk} : k = 1, 2, \dots, M_{ij}^{(1)}\}, \forall i, j.$$

Finally r_{ijk} contains a number of pixels in level 0. In this way, we have three levels of adjacency graphs,

$G^{(\sigma)} = \langle V^{(\sigma)}, E^{(\sigma)} \rangle$, $\sigma = 1, 2, 3$.
A vertex $v \in V^{(\sigma)}$ corresponds to a subset of vertices at $V^{(\sigma-1)}$.

We formulate the problem as graph partition in the Bayesian framework. Let $\mathbf{I}_1, \mathbf{I}_2$ be two consecutive images in a motion sequence, and $G = \langle V, E \rangle = G^{(1)}$ the image lattice. For computational efficiency, for each pixel $(x, y) \in V$ we discretize its velocity (dx, dy) to an $m \times m$ grid. For example, if we assume the maximum speed is 3 pixels with 1/2 pixel accuracy, we have 13×13 possible velocity. Thus each pixel has a composite label $\ell = (i, j, k = (dx, dy))$ indexing the object O_i , image region R_{ij} and its velocity r_{ijk} respectively. Thus a partition π represents, in a hierarchical structure, a velocity field \vec{V} , an image segmentation R and a motion segmentation O . We maintain an accretion map A such that $A(\vec{x}) = 0$ represents the accreted pixels (that were occluded in the first frame), and model them by the image models H_{ij} , not by the intensity models H_{ij}^p of prediction by motion. We assume the pixels in A are independent and we can assign each pixel $A(\vec{x}) = 0$ if and only if $P(A(\vec{x}) = 0)/P(A(\vec{x}) \neq 0) > 1$.

The graphs at each level are updated online, at each move. For level 1 we maintain a "graph sparse histogram" which is a hash table containing the pairs of atomic region indices which have non-zero boundary, and the boundary length. When the boundary length becomes zero, the entry is erased from the table. The other levels have much fewer nodes and they can be updated online, with or without hash table. There is a lot of overhead in maintaining the graphs, especially the atomic region graph. An alternative would be to update the whole graph more rarely and run more steps at the higher levels consecutively, but we didn't try this yet.

The posterior probability is defined as $p(\pi|\mathbf{I}_1, \mathbf{I}_2; \theta) \propto p(\mathbf{I}_2|\mathbf{I}_1, \pi, \theta)p(\pi, \theta)$. More exactly

$$P(I_2 | I_1, \pi, \theta, \{H_{ij}^p\}, A) = \prod_{i,j} \prod_{\vec{x} \in R_{ij}; A(\vec{x}) \neq 0} P_{H_{ij}^p}(I_2(\vec{x}) - I_1(\vec{x} - \vec{v}(\vec{x}))) \cdot \prod_{i,j} \prod_{\vec{x}; A(\vec{x})=0, \vec{x} \in R_{ij}} p_{H_{ij}}(I_2(\vec{x})) \quad (10)$$

where $\vec{v}(\vec{x})$ is the velocity of pixel \vec{x} , as obtained from the current pixel label in the partition π .

The prior $p(\pi, \theta) = p(\pi|\theta)p(\theta)$ assumes a uniform probability on the motion parameters θ_i and on the image models H_{ij} and H_{ij}^p and has a MRF probability on the velocity field and a Gaussian probability to measure the deformation of the velocity field from the rigid motion model behind.

$$p(\pi|\theta) \propto \prod_i \exp \left(-\alpha \sum_{\vec{x} \in O_i} \left[(1/2\sigma_i^2) |\vec{v}(\vec{x}) - \vec{v}_i(\vec{x})|^2 + \beta \sum_{\vec{x}' \in \partial \vec{x}} (|\vec{v}(\vec{x}') - \vec{v}(\vec{x})|) \right] \right) \prod_{i,j} e^{-\gamma \text{len}(\partial R_{ij})} \prod_{i,j,k} e^{-\delta \text{len}(\partial r_{ijk}) - \eta \text{var}(r_{ijk})} \quad (11)$$

The SW graph edges at the pixel level between two pixels i, j are obtained by finding the best motion $\vec{v} = (u, v)$ that fits both pixels at the same time. Then

$$q(i, j) = 0.1 + 0.9 * e^{-d(i, j, \vec{v})/7 - |I_2(i) - I_2(j)|/10} \quad (12)$$

$$d(i, j, \vec{v}) = |I_2(i) - I_1(i - \vec{v})| + |I_2(j) - I_1(j - \vec{v})| \quad (13)$$

At the atomic region level, the edge weights between two adjacent atomic regions is based on the KL divergence between the intensity histograms, as in [1].

At the region level, we maintain 2d motion histograms for each region (histograms of the motion (u, v) of all the pixels of a region). Then the edge weight between adjacent regions is based on the KL divergence between these histograms.

Given the posterior probability $p(\pi)$ above, it is straightforward to run the various SW-cuts sampling $p(\pi)$. For clarity, we omit the details in estimating the model parameters θ .

4.2. Experiments

We run the multigrid and multi-level SW-cut on a number of synthetic real world motion images. We show four results in Fig.6. The first image shows two moving rectangles – a case discussed in [17], where only the 8 corners provide reliable local velocity (aperture problem) and the image segmentation is instrumental in deriving the right result. For the other three sequences, the SW-cuts obtain satisfactory results despite large motion and complex background. The cheetah image in Fig.1 is another example.

For comparison of the different SW-cut sampling methods, we plot in Fig.7, the energy vs time of the SW-cut multi-grid and multi-level presented in this paper, for a simple image segmentation problem (histogram intensity model with boundary length prior) on two levels (atomic regions of constant intensity and image segmentation). We also plot the behavior of the SW-cut algorithm running on atomic regions obtained from edge detection, and on a big random window, comparable to the whole image (almost like the SW-cut working on all the pixels in the image at the same time, but slightly more efficient). The initial partition $\pi(0)$ is random.

Even though the SW-cut on atomic regions is extremely fast (5 sec), it cannot reach the same energy minimum as the other methods, because the atomic regions cannot be broken. The multigrid method is the fastest among the methods that work directly on pixels. On the other hand, the convergence behaviour of the multi-level SW-cut is almost linear. Even though it seems to be slower than the multi-grid, actually the energy level attained with the multi-level after 600s is slightly smaller than all of the other methods.

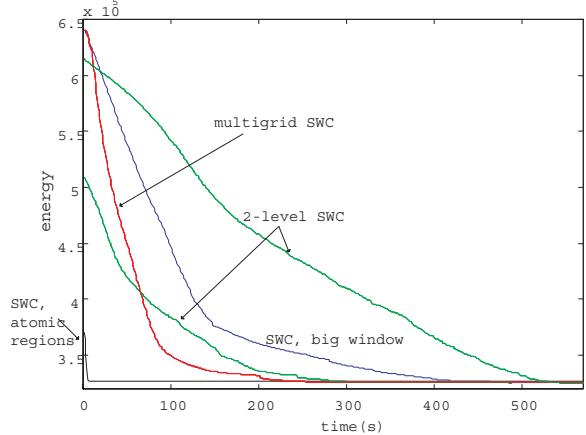


Figure 7. Convergence comparison of multi-grid and multi-level SW-cuts. (see text for explanation)

5. Future work

In an ongoing project, we are extending the SW-cuts to segmenting long motion sequences where the adjacency graph is 3-dimensional with spatial and temporal relations. Thus each step of the SW-cut will flip the labels of a 3D subgraph (volume). In this way, we shall segment and track the motion both forward and backward in time for optimal Bayesian solutions. At the same time, we try to improve the motion priors to be better suited for our hierarchical representation and computation approach.

References

- [1] A. Barbu, S.C. Zhu, "Graph partition by Swendsen-Wang Cuts", *ICCV*, Nice, France, 2003.
- [2] E. Bienenstock, S. Geman, D. Potter, "Compositionality, MDL, and object recognition", *NIPS*, 1997.
- [3] M. J. Black, A. D. Jepson, "Estimating optical flow in segmented images using variable order parametric models with local deformations", *IEEE Trans PAMI*, Vol. 18, no. 10, pp. 972–986, 1996.
- [4] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts. *PAMI* 2001
- [5] C. Cooper, A.M. Frieze, "Mixing Properties of the Swendsen-Wang Process on Classes of Graphs", *Random Structures and Algorithms* 15, 242-261, 1999.
- [6] W. T. Freeman, E.C. Pasztor, O.T. Carmichael, Learning low-level vision. *IJCV* 2000

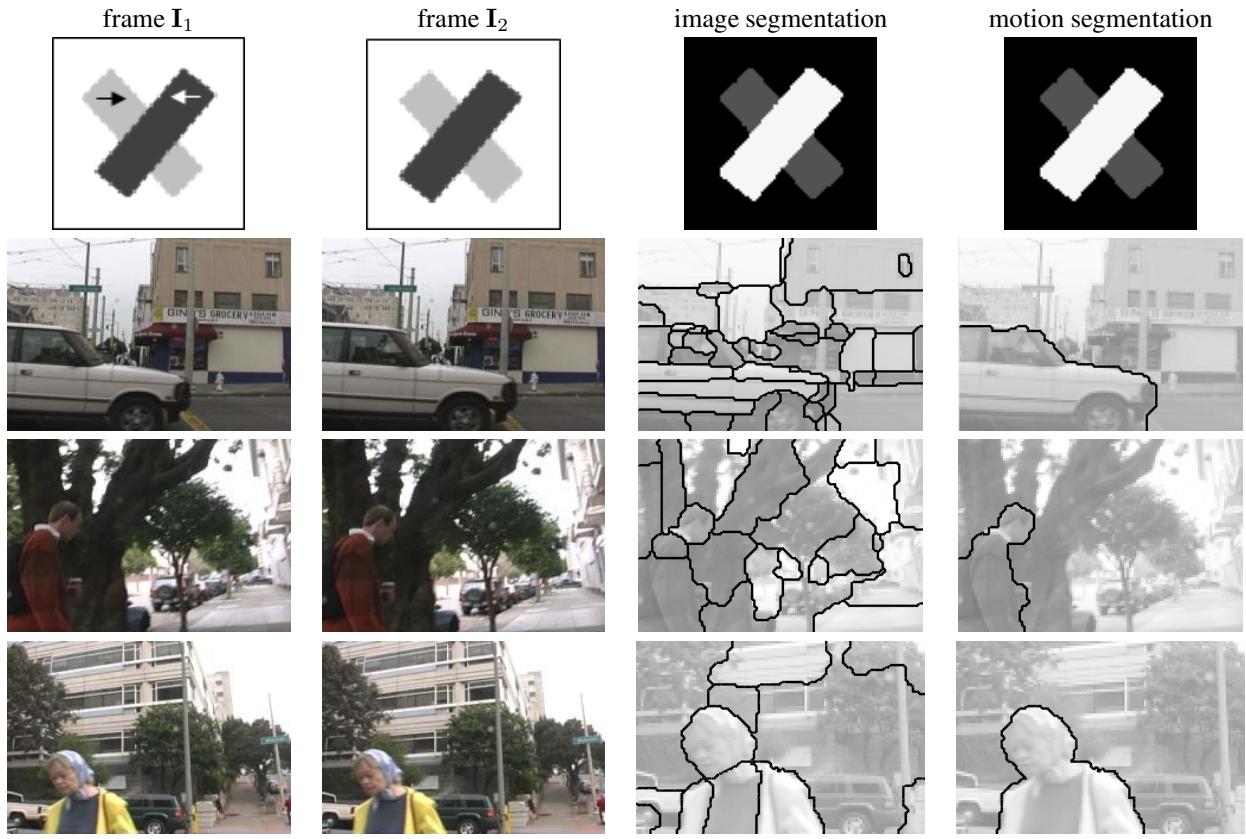


Figure 6. Hierarchical image and motion segmentation at 3 levels. From left to right: first frame, second frame, image segmentation, motion segmentation. The image segmentation is the result at level 2 and the motion segmentation is the result at level 3.

- [7] S. Geman, D. Geman, “Stochastic relaxation, Gibbs distrib., Bayesian restoration of img”, *PAMI*, vol. 6, 721-741, 1984.
- [8] R. Hummel, S. Zucker, “On the foundations of relaxation labelling processes”, *IEEE Trans. PAMI*, vol. 3, no 5, 1983.
- [9] V. Kolmogorov, R. Zabih, “What energy functions can be min. via graph cuts?”, *ECCV*, 2002
- [10] J.S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer-Verlag, NY, Inc, 2001 (page 134).
- [11] N. Metropolis, et al. “Eqns of the state calculation by fast computing machines”, *J. Chem. Phys.*, 21, 1087-91, 1953.
- [12] P.H.S. Torr, “Geometric motion segmentation and model selection”, *Philosophical Trans of the Royal Society A*, pp. 1321–1340, 1998.
- [13] S. Roy, I. Cox, “A max-flow formulation of the n-camera stereo correspondence problem”, *ICCV*, 1998.
- [14] J. Shi, J. Malik, “Normalized cuts and image segmentation”, *PAMI*, **22**, no 8, pp. 888-905, 2000.
- [15] Z.W. Tu, S. C. Zhu, “Image segmentation by data-driven MCMC”, *PAMI*, **24**, no. 5, 2002.
- [16] R.H. Swendsen, J. S. Wang, “Nonuniversal critical dynamics in MC simulations”, *Phys. Rev. Lett.*, **58** no. 2, pp.86-88, 1987
- [17] Y. Weiss, “Smoothness in Layers: Motion segmentation using nonparametric mixture estimation”, *Proc. of IEEE CVPR* pp. 520-527, 1997.
- [18] A.L. Yuille, N.M. Grzywacz, “A mathematical analysis of the motion coherence theory” *Int. J. Computer Vision* **3**, pp. 155–175, 1989.