Submitted to Int'l J. of Computer Vision, October 2003. First Review March 2005. Revised June 2005.
.

Parsing Images into Regions, Curves, and Curve Groups

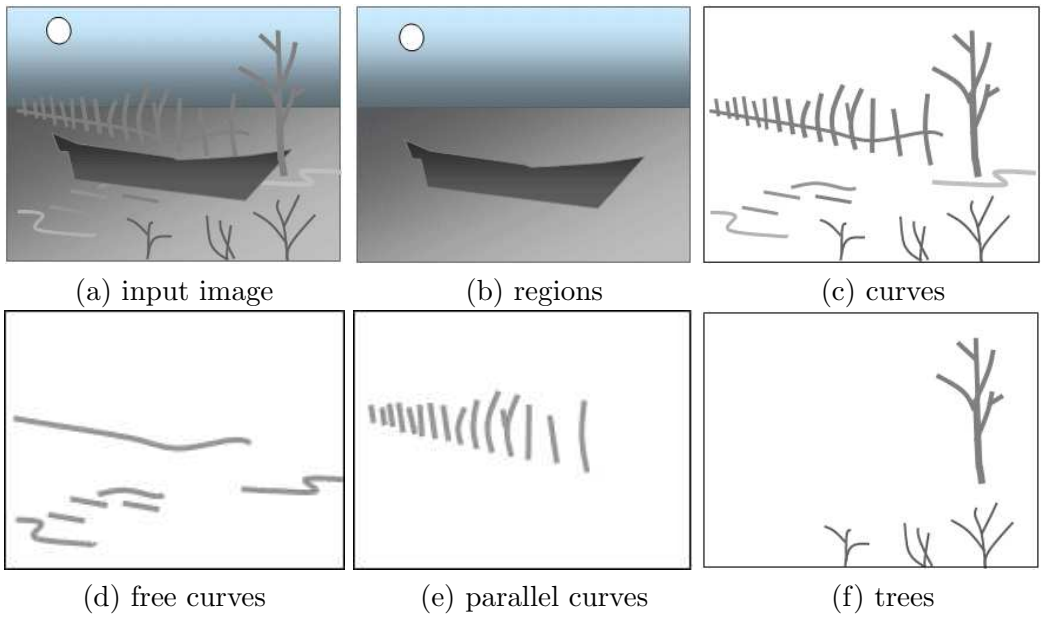
Zhuowen Tu[1] and Song-Chun Zhu[1,2]

Departments of Statistics[1] and Computer Science[1,2],
University of California, Los Angeles,
Los Angeles, CA 90095. emails: {ztu,sczhu}@stat.ucla.edu

Abstract

In this paper, we present an algorithm for parsing natural images into middle level vision representations – regions, curves, and curve groups (parallel curves and trees). The algorithm is targeted for an integrated solution to image segmentation and curve grouping through Bayesian inference. The paper makes the following contributions to the literature. (1) It studies a layered (or 2.1D-sketch) representation integrating both regions and curves models which compete to explain an input image in terms of maximizing a Bayesian Posterior probability. The curve layer occludes the region layer and the curves observe a partial order occlusion relation among themselves. (2) It studies a Markov chain search scheme which consists of many pairs of reversible jumps to traverse the complex solution space. Six Markov chain jumps are designed as *Metropolized Gibbs Samplers* (MGS). A MGS proposes the next state within the jump scope of the current state according to a conditional probability like a Gibbs sampler and then accepts the proposal with a Metropolis-Hasting step. One pair of jumps is designed by the Swendsen-Wang cut method for curve grouping. The paper discusses systematic design strategies of devising reversible jumps for the complex inference task. (3) The proposal probability ratios in jumps are factorized into ratios of discriminative probabilities. The latter are computed in a bottom-up process and drive the Markov chain dynamics in a Data-Driven Markov Chain Monte Carlo framework. We demonstrate the performance of the algorithm in experiments with a number of natural images.

Keywords: Image Segmentation, Perceptual Organization, Curve Grouping, Graph Partition, Data-Driven Markov Chain Monte Carlo, Metropolized Gibbs Sampler.



(a) input image (b) regions (c) curves

(d) free curves (e) parallel curves (f) trees

Figure 1: An illustration of parsing an image into regions, curves, and curve groups. (a) is an input image which is decomposed into two layers – (b) a layer of regions and (c) a layer of curves. The curves are further divided into (d) free curves, (e) a parallel curve group for the fence, and (f) the trees. The curves observe a partial order occlusion relation.

1 Introduction

1.1 Objectives and contributions

In this paper, we present an algorithm for parsing natural images into middle-level vision representations – regions, curves, and curve groups (parallel curves and trees). The algorithm is targeted for an integrated solution to image segmentation and curve grouping through Bayesian inference. We adopt a generative model in a layered (or 2.1D sketch) [30, 40] representation illustrated in Fig. 1. A region is a two-dimensional compact area with coherent intensities. We specify two types of coherence for regions, one is constant intensity with homogeneous texture, and the other is smooth shading. Each type of coherence is specified by a family of probability models. A curve is one-dimensional shape structure with smooth intensity profile at the cross section and along the curve. It may be considered as a degenerated region. This is different from other works which refer curves to the boundaries of 2D regions [23, 22, 27]. We are interested in three types of curve structures in the paper. (1) *free curves* – independent and elongated 1D structures. (2) *parallel groups* –curves that form a 1D Markov chain structure along their normal directions, such as railing

and zebra stripes, and (3) trees – curves arranged Markov tree structures. All curve structures are assumed to observe a partial-order occlusion relation and they all occlude the region layer.

While there are a wealthy body of work about *image segmentation* and *curve detection* and *grouping* respectively, the two problems have not been studied together in explicit representations. The integration is important for achieving improved results in either tasks since they jointly explain the input image. On one side, conventional segmentation algorithms assume that images consist of two-dimensional compact regions and thus produce degenerated results when they encounter one-dimensional curve objects. For example, Fig. 2 shows a few examples of image segmentation using a Data-Driven Markov Chain Monte Carlo (DDMCMC) method [37] and the curves make the segmentation rather cluttered. For comparison Figures 14-23 demonstrate much improved results when the curve structures are represented and computed separately. On the other side, for lack of image models for background regions, curve detection and grouping algorithms often assume uniform background, or as an alternative, they have to adopt discriminative curve models that work on the differences between curves and background.

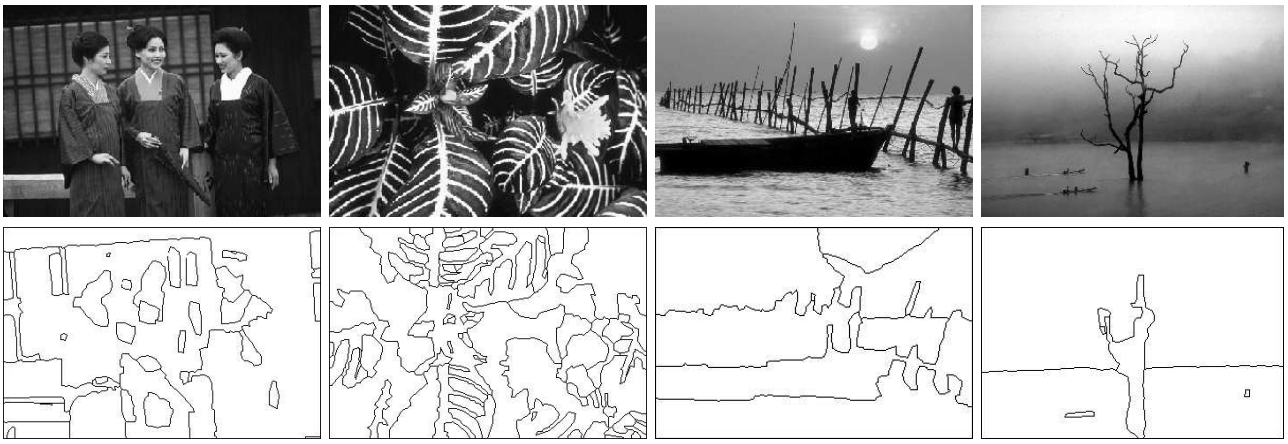


Figure 2: Degraded results in image segmentation in the presence of curves. In comparison, Figures 14-23 show much improved results when the curve structures are represented and computed.

Given an input image, our objective is to infer an unknown number of regions, free curves, and parallel groups and trees, with occlusion relation recovered and their probability models selected and fitted – all in the process of maximizing (or simulating) a Bayesian posterior probability. The algorithm searches for optimal solutions in a complex state space which contains a large number of

subspaces of varying dimensions for the possible combinations of regions, curves, and curve groups.

The paper is mainly focused on studying a systematic search strategy in such complex state spaces by Markov chain Monte Carlo (MCMC) methods. The Markov chain simulates (or draw samples) from a posterior probability. The following are three basic considerations in our MCMC design.

Firstly, the Markov chain should be irreducible so that it can traverse the entire solution space. Our Markov chain consists of six pairs of "simple jumps" and one pair of "composite jumps" for operating on the curves, together with a number of reversible jumps on the regions which are referred to previous image segmentation paper [37]. These pairs of jumps are selected probabilistically and the resulting Markov chain has a transition graph which connects all states in the search space. The Markov chain can reach any states from an arbitrary initialization.

Secondly, each simple jump operates on $1-2$ curves or curve elements. We study the scopes of the jumps within which the algorithm proposes the next state according to a conditional probability like a Gibbs sampler. The proposal is then accepted in a Metropolis-Hastings step. This is why it is called *Metropolized Gibbs Sampler* (MGS [25]). A composite jump can operate on a selected set of curves through a cluster sampling method (the Swendsen-Wang cut [2, 3]).

Thirdly, the computational cost at each jump step should be small. The proposal probability ratios in our design are factorized and computed by discriminative probability ratios. These discriminative probabilities are computed in bottom-up processes which are then used to activate the generative models in a top-down process. As Fig. 12 illustrates, each simple jump maintains a list of "particles" which are weighted hypotheses with the weights expressing the discriminative probability ratios. Then a particle is proposed at a probability proportional to its weight within the list (scope). The higher the weight is, the more likely a particle will be chosen. In contrast, the composite jump introduces auxiliary variables in the proposal probability and explores a combinatorial number of candidates (clusters) instead of maintaining a list explicitly.

1.2 Relation to previous work and alternative methods

1. Relation to previous work on Markov chain Monte Carlo. Stochastic computing with reversible jumps was pioneered in [19, 18], and the jump-diffusion method was used successfully in [24, 35] for automatic target recognition and pose estimation. A Data-Driven Markov Chain Monte Carlo

(DDMCMC) framework was proposed to improve the speed of reversible jumps by computing the proposal probabilities with factorized discriminative models. The framework has been applied to object detection [42], image segmentation [37] and range segmentation [12]. Lately, the DDMCMC framework has been applied to integrating high level vision tasks such as face and text detection with the segmentation process [39]. This current work has been extended in [20] to 3D scene reconstruction from a single image based on the region and curve representation. The work is part of a series of work for achieving an integrated vision inference engine for segmentation, grouping, and object recognition. In this paper we shall solely focus on the curve grouping aspect and its interactions with image segmentation.

In recent years, the Markov chain Monte Carlo methods have attracted considerable interests in vision and demonstrated computational power in traditional tasks such as structure from motion [11, 14] and 3D modeling[13]. There is a growing need for systematic ways of designing and analyzing effective Markov chain searches in complex vision tasks. In this paper we intend to provide detailed descriptions in an effort to make the MCMC designs simple and easy to use.

2. Relation to the other work in curve detection and grouping.. Curve detection and tracing have been extensively studied in several areas. For example, active contours (SNAKE) [23], road tracing in satellite images [16], medical image analysis [43], object tracking [22], and image coding using ridgelets and curvelets [6, 7]. Existing methods have various restrictive assumptions. (1) Many methods require manual initialization of the curve near the right position [23, 22, 10] or manually initializing the starting point for tracing [16]. (2) Most algorithms assume uniform background [1, 43] for the lack of generative models for the background regions. The popular SNAKE and active contour models use a discriminative representation that works on the difference between the curves and background rather than generative models for images. (3) Image coding algorithms [6, 7] assume a generative model that images are linear addition of curve elements/bases. Our early attempt [38] adopted this additive model with the image bases organized in a Markov chain. The additive model results in artifacts, such as blurry boundaries, and much improved results are observed using occlusion model in this paper. (4) In vision, many perceptual grouping methods work on edge maps rather than the original images. We argue that generative image models are needed for recovering from errors in the edge detection stage and for interacting with other types of objects in images.

3. Comparison with alternative methods. The Markov chain jumps can be considered as generalizations to conventional gradient descent moves in three aspects. (I) A jump can change dimensions in the state space by changing the number of objects, while gradient methods only move in spaces of fixed dimensions. (II) A jump can move in a rather large scope at a single step, while gradient methods move within a small nearest neighborhood. (III) A jump samples the next state probabilistically in its scope, while gradient methods make a greedy decision.

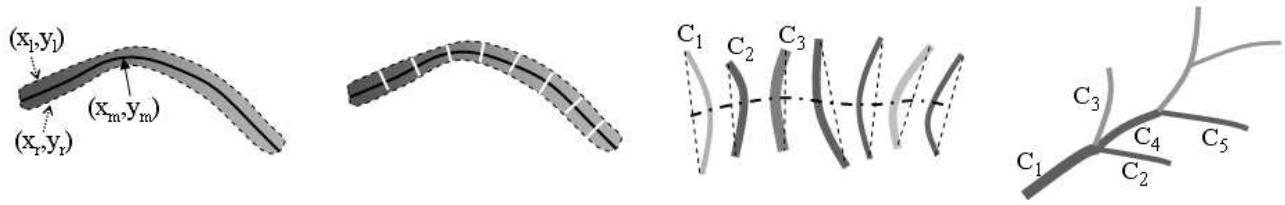
1.3 Organization of the paper

In Section (2) we first present the generative models for regions, curves, and curve groups, and formulate the problem as Bayesian inference. Then we present the algorithm in four sections. In Section (3) we discuss the basic principles of Metropolized Gibbs Sampler (MGS) methods, speed analysis, and strategies for good designs. In Section (4) we study reversible jumps for structural and occlusion relation changes involving regions and free curves. Then we show a series of experiments in Section (5) and conclude the paper with a discussion in Section (6).

2 Generative models and Bayesian formulation

In this section, we present generative models for both regions and curve structures, and formulate the inference problem in a Bayesian framework.

2.1 Generative models of curves



(a) free curve (b) discretized curve (c) parallel group (d) tree

Figure 3: Representations of curves and curve groups. (a) A free curve in continuous representation. (b) A free curve is discretized into a chain of "bars". (c) Curves for a parallel group, (d). Curves for a Markov tree.

In this paper, we consider three types of curve models illustrated in Fig. 3.

1. Free curves. A free curve, denoted by C, is represented by its medial axis $\mathbf{c}_m(s) = (x_m(s), y_m(s))$

and its width $2w(s)$ for $s = [0, L]$. L is the curve length. In a continuous representation, a free curve C occupies an elongated area or domain $\mathcal{D}(C)$ bounded by the left and right side boundaries denoted respectively by $\mathbf{c}_l(s) = (x_l(s), y_l(s))$ and $\mathbf{c}_r = (x_r(s), y_r(s))$. Fig. 3.a shows the boundaries in dashed lines.

$$\mathbf{c}_l(s) = \mathbf{c}_m(s) - w(s)\mathbf{n}(s), \quad \mathbf{c}_r(s) = \mathbf{c}_m(s) + w(s)\mathbf{n}(s), \tag{1}$$

where $\mathbf{n}(s)$ is the unit normal of $\mathbf{c}_m(s)$. Intuitively, a curve is a degenerated region parameterized by a 1D axis. Usually $w(s)$ is only $1 - 3$ pixels wide and $w << L$. This causes major topology problems in image segmentation where the two boundaries $\mathbf{c}_l(s)$ and $\mathbf{c}_r(s)$ could often intersect and generate numerous trivial regions. This problem will be resolved with the 1D representation. The intensities of the curve often exhibit a globally smooth shading patterns, for example, the stripes in a zebra image. Thus we adopt a quadratic function for curve intensities,

$$\mathbf{J}(x, y; \theta_0) = ax^2 + bxy + cy^2 + dx + ey + f, \quad (x, y) \in \mathcal{D}(C), \tag{2}$$

with parameters $\theta_0 = (a, b, c, d, e, f)$. Therefore, a free curve is described by the following variables in continuous representation

$$C = (L,\ \mathbf{c}_m(s)_{s=0}^{L},\ w(s)_{s=0}^{L},\ \ \theta_0, \sigma).$$

where σ is the variance of the intensity noise (residue).

While this continuous representation is convenient for deriving diffusion equations (PDEs), we should also work on a discrete representation in other sections where the domain $\mathcal{D}(C)$ is a set of pixels in lattice and C is a chain of elongated bars as Fig. 3.b illustrates.

The prior model for $p(C)$ prefers smooth medial axis, narrow and uniform width, and it also has a term for the area of the curve in order to match with the region prior.

$$p(C) \propto p(\mathcal{D}(C))p(\mathbf{c}(s))p(w(s)) \propto e^{-E(C)}. \tag{3}$$

The energy $E(C)$ is the sum of three terms

$$E(C) = \gamma_c|\mathcal{D}(C)|^{\rho} + \lambda L + E_o(w), \tag{4}$$

where ρ, λ are constants and are fixed in our experiments, and γ_c is a scale factor that can be adjusted to control the number of curves. $E_o(w)$ is a term which constrains width $w(s)$ to be small.

We denote the intensities inside the curve domain by $\mathbf{I}_{\mathcal{D}(C)}$, and assume the reconstruction residue follows iid Gaussian $\mathcal{N}(0;\sigma^2)$, so the image likelihood is

$$p(\mathbf{I}_{\mathcal{D}(C)}|C) = \prod_{(x,y)\in\mathcal{D}(C)} \mathcal{N}(\mathbf{I}(x,y) - \mathbf{J}(x,y;\theta_0);\, \sigma^2). \tag{5}$$

2. Parallel curve groups. A parallel curve group consists of a number of n nearly parallel curves as Fig. 3.c shows. Each curve $C_i, i = 1, 2, ..., n$ is summarized by a short line segment connecting its end points (see the dashed lines in Fig. 3.c). They represent curve structures, such as zebra strips, grids, and railings in natural images. Grouping curves into a parallel group is encouraged in the model as it reduces coding length and it is crucial for perceiving the whole object from its parts. For example, separating a zebra from the herd. We denote a parallel curve group by

$$pg = (n, \{C_1, C_2, ..., C_n\}, \{\alpha_1, \alpha_2, ..., \alpha_n\}),$$

$\alpha_i \in \{1, ..., n\}$ is the index to the curve preceding C_i in the chain.

The prior model for a pg is a first order Markov model in a Gibbs form with a singleton energy on individual curve and a pair energy for two consecutive curves

$$p(pg) \propto \exp\{-\lambda_0 n - \sum_{i=1}^{n} E(C_i) - \sum_{i=2}^{n} E_{pg}(C_i, C_{\alpha_i})\}. \tag{6}$$

The singleton $E(C_i)$ is inherited from the free curve model. For the pair energy, we summarize each curve C_i by five attributes: center (x_i, y_i), orientation θ_i of its associate line-segment, and length L_i of the line segment, curve average width (thickness) $\bar{w}_i$, and average intensity μ_i. $E_{pg}(C_i, C_{\alpha_i})$ measures the differences between these attributes.

3. Markov trees. Fig. 3.d shows a number of curves in a Markov tree structure. We denoted it by

$$T = (n, \{C_1, C_2, ..., C_n\}, \{\beta_1, \beta_2, ..., \beta_n\}).$$

$\beta_i \in \{1, ..., n\}$ is the index to the parent curve of C_i. Thus the prior probability is

$$p(T) \propto \exp\{-\lambda_0 n - \sum_{i=1}^{n} E(C_i) - \sum_{\alpha_i \neq \emptyset} E_T(C_i, C_{\beta_i})\}. \tag{7}$$

Again, $E(C_i)$ is inherited from the free curve. The term for C_i and its parent C_{α_i}, $E_T(C_i, C_{\alpha_i})$, measures the compatibility such as end-point gap, orientation continuity, thickness, and intensity between the parent and child curves.

The parallel group pg and tree T inherit the areas from the free curve, thus

$$\mathcal{D}(pg), \mathcal{D}(T) = \cup_{i=1}^{n} \mathcal{D}(C_i). \tag{8}$$

It also inherits the intensity function $\mathbf{J}(x, y; \theta_i)$ from each free curve $C_i, i = 1, 2, ..., n$. In summary, the intensity models for C, pg, T are all generative for image $\mathbf{I}$ as

$$\mathbf{I}(x, y) = \mathbf{J}(x, y; \theta) + \mathcal{N}(0; \sigma^2), \;\; (x, y) \in \mathcal{D}(C_i), \, \mathcal{D}(pg), \text{ or } \mathcal{D}(T). \tag{9}$$

2.2 Generative models of regions

Once the curves explain away the elongated patterns, what left within each image region in the background are often modeled reasonably well with simple region models, except high level objects which will be studied in [39]. In this paper, we adopt two simple region models in comparison to the four models in [37] We denote by $R \subset \Lambda$ a 2D region and $\mathbf{I}_R$ the intensities inside R.

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

$$\mathbf{J}(x, y; 1, \theta) = \mu, \;\; \mathbf{I}(x, y) = \mathbf{J}(x, y) + \eta, \;\; \eta \sim \mathcal{H}, \;\; (x, y) \in R.$$

With a slight abuse of notation, we denote by $\theta = (\mu, \mathcal{H})$ the parameters used in a region.

The second model assumes a 2D Bezier spline function with additive noise. The spline accounts for global smooth shadings.

$$\mathbf{J}(x, y; 2, \theta) = B'(x) M B(y), \;\; \mathbf{I}(x, y) = \mathbf{J}(x, y; \theta_2) + \eta, \;\; \eta \sim \mathcal{H}, \;\; (x, y) \in R.$$

where $B(x) = ((1-x)^3, 3x(1-x)^2, 3x^2(1-x), x^3)$ is the basis and M is a 4×4 control matrix. The parameters are $\theta = (M, \mathcal{H})$ as used in [37]. The likelihood probability is

$$p(\mathbf{I}_R | R, \theta) \propto \prod_{(x,y) \in \mathcal{D}(R)} \mathcal{H}(\mathbf{I}(x, y) - \mathbf{J}(x, y; \ell, \theta)), \quad \ell \in \{1, 2\}. \tag{10}$$

The prior for a region R assumes short boundary length ∂R (smoothness) and compact area $|\mathcal{D}(R)|$,

$$p(R) \propto \exp\{-\gamma_r |\mathcal{D}(R)|^{\rho} - \frac{1}{2} \lambda |\partial R|\}, \tag{11}$$

where ρ and λ are constants that are fixed for all the experiments in this paper, and γ_r is a scale factor that can be adjusted to control the number of regions in the segmentation.

2.3 Occlusion, partial order relation, and partition of lattice Λ

We collect all the curves, including free curves and the curves in the parallel groups and trees, in a set $A = (C_1, C_2, ..., C_N)$, and define a partially ordered set (*poset* [33]) $\mathcal{PR} = \langle A, \preceq \rangle$. $b \preceq c$ means that curve b occludes curve c or b is on top of c. $\mathcal{PR}$ is represented by a directed acyclic graph called Hasse diagram. Figure 4 shows an example of the Hasse diagram for $\mathcal{PR} = \{\langle a, b\rangle, \langle b, d\rangle, \langle a, d\rangle, \langle a, c\rangle, \langle c, d\rangle, \langle e, f\rangle\}$ on a set $A = \{a, b, c, d, e, f\}$.

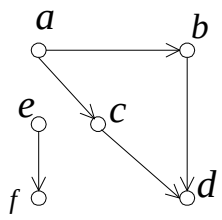


Figure 4: A Hasse diagram for a partial order relation

As we define curves as the basic elements in A, the curves in a parallel group or a tree can occlude each other. By default, all curves in A occlude the region layer. It is worth mentioning that the occlusion representation is important for producing improved results over the additive representation in our previous experiments [38] that generate images by superimposing image bases. The occlusion relation between two curves are often evident at the T-junctions or cross-junctions in images. Figure 7.(c) shows an example of junction and corner detection as bottom-up computation for the occlusion points.

The occlusion relation $\mathcal{PR}$ forms a partition of the 2D discrete lattice Λ. Each curve C occupies pixels in its domain $\mathcal{D}(C)$ minus the pixels covered by other curves occluding C,

$$\Lambda_C = \mathcal{D}(C) - \cup_{C' \preceq C} \mathcal{D}(C'). \tag{12}$$

Therefore the domains for parallel groups and trees are respectively

$$\Lambda_{pg} = \cup_{C \in pg} \Lambda_C \quad \text{and} \quad \Lambda_T = \cup_{C \in T} \Lambda_C. \tag{13}$$

The visible part of a region R is $\Lambda_R = \mathcal{D}(R) - \cup_{C \in A} \Lambda_C$. [1]

[1] For clarity of notation, we use $\mathcal{D}(C)$, $\mathcal{D}(pg)$, $\mathcal{D}(T)$, and $\mathcal{D}(R)$ in equations (3), (4), (5), (9), (11), and (10). They are actually replaced by Λ_C,Λ_{pg},Λ_T,Λ_R respectively in implementation.

2.4 Bayesian formulation for probabilistic inference

Given an image $\mathbf{I}$, our objective is to compute a representation of the scene (world W) in terms of a number of regions W^r, free curves W^c, parallel curve groups W^{pg}, trees W^t, and a partial order $\mathcal{PR}$. We denote the representation by variables

$$W = (W^r, W^c, W^{pg}, W^t, \mathcal{PR}).$$

The region representation W^r includes the number of regions K^r, and each region R_i has a label $\ell_i \in \{1, 2\}$ and parameter θ_i for its intensity model

$$W^r = (K^r, \{(R_i, \ell_i, \theta_i) : i = 1, 2, ..., K^r\}).$$

Similarly, we have $W^c = (K^c, C_1, ..., C_{K^c})$, $W^{pg} = (K^{pg}, pg_1, pg_2, ..., pg_{K^{pg}})$, and $W^t = (K^t, T_1, T_2, ..., T_{K^t})$. The problem is posed as Bayesian inference in a solution space Ω.

$$W^* = \arg\max_{W \in \Omega} p(\mathbf{I}|W)p(W).$$

By assuming mutual independence between W^r, W^c, W^{pg}, W^t we have the prior model

$$p(W) = \Big(p(K^r)\prod_{i=1}^{K^r} p(R_i)\Big)\Big(p(K^c)\prod_{i=1}^{K^c} p(C_i)\Big)\Big(p(K^{pg})\prod_{i=1}^{K^{pg}} p(pg_i)\Big)\Big(p(K^t)\prod_{i=1}^{K^t} p(T_i)\Big). \quad (14)$$

The prior for individual $p(R), p(C), p(pg), p(T)$ are given in previous subsections.

As there are N curves in total including the free curves, and curves in the parallel groups and trees, thus the likelihood model follows the lattice partition and eqns (5) and (10).

$$p(\mathbf{I}|W) = \prod_{i=1}^{K^r}\prod_{(x,y)\in\Lambda_{R_i}} \mathcal{H}((\mathbf{I}(x,y) - \mathbf{J}(x,y;\ell_i,\theta_i)) \cdot \prod_{j=1}^{N}\prod_{(x,y)\in\Lambda_{C_j}} \mathcal{N}((\mathbf{I}(x,y) - \mathbf{J}(x,y;\theta_j);\sigma_j^2). \quad (15)$$

Since all objects use generative model for reconstructing $\mathbf{I}$, these models are directly comparable and they compete to explain the image. This property is crucial for the integration of region segmentation and curve grouping.

Our goal is to design an algorithm to make inference of the W^* which maximizes the posterior $p(W|\mathbf{I})$ by sampling W in the solution space with a fast simulated annealing procedure. This poses rather serious challenges even though we have simplified the image models above. The main difficulty is to deal with objects with different structures and explore a large number of possible combinations of regions, curves, parallel groups and trees in an image. Especially our objective

is to achieve automatic (i.e. no assumption for manual initialization) and nearly globally optimal solutions.

We present the algorithm in the next sections. Limited by space, we only present novel components for solving problems arising in the integration of segmentation and grouping. Readers are referred to [37] for details of image segmentation by DDMCMC. We focus on the analysis and design of Metropolized Gibbs Sampler (MGS) and its approximation followed by the detailed discussion of the reversible jumps for structural and occlusion relation changes involving regions, free curves, and curve groups.

3 Searching complex solution space by Markov chain

A main technical challenge in the integrated image parsing problem is that we must infer an unknown number of objects – regions, free curves, and curve groups, with their occlusion relations computed and their probability models selected and fitted. The algorithm must search for the optimal solution in space Ω which consists of a large number of subspaces of varying dimensions. In this section, we overview the basic concepts, principles, and speed criteria for designing Markov chains that can traverse the solution space.

3.1 Designing reversible jumps

In this subsection, we shall focus on the essential practical problems in designing the reversible jumps for exploring the space Ω.

3.1.1 Overview of MCMC design

Our goal is to maximize a posteriori (MAP) probability $p(W|\mathbf{I})$ in solution space Ω, Note that W has both discrete and continuous random variables and both can be sampled with the reversible jumps[18]. In practice, diffusion processes are added to the reversible jumps[19, 24, 35, 12] for effective computation of some continuous variables, such as the boundary of regions. In this paper, we shall focus on the reversible jumps for clarity and we omit the region competition processes for boundary diffusion [41].

The sampling algorithm simulates a Markov chain denoted by a triplet $\mathcal{MC} =< \Omega, \nu, \mathbf{K} >$. $\nu(W_o)$ is the probability for the initial state W_o at time $t = 0$, and $\mathbf{K}(W_A, W_B)$ denotes the transition probability from state W_A to state W_B for any $W_A, W_B \in \Omega$. The kernel $\mathbf{K}$ shall have a

unique stationary probability $p(W|\mathbf{I})$, i.e.

$$\sum_{W_A \in \Omega} p(W_A|\mathbf{I})\mathbf{K}(W_A, W_B) = p(W_B|\mathbf{I}), \quad \forall\, W_B \in \Omega. \tag{16}$$

In practice, the requirement is replaced by a stronger condition – the detailed balance equation,

$$p(W_A|\mathbf{I})\mathbf{K}(W_A, W_B) = p(W_B|\mathbf{I})\mathbf{K}(W_B, W_A), \quad \forall\, W_B \neq W_A. \tag{17}$$

If we write ν and $\mathbf{K}$ as a row vector and a matrix respectively. The Markov chain visits states $W_o, W_1, ..., W_t$ sequentially. At time t the Markov chain state W_t follows probability

$$W_t \sim p_t(W) = \sum_{W_o} \nu(W_o)\mathbf{K}^t(W_o, W). \tag{18}$$

As t increases, the Markov chain probability $p_t(W)$ shall approach $p(W|\mathbf{I})$, then the samples W_t approximately follows the posterior probability $p(W|\mathbf{I})$. The convergence of $p_t(W)$ to the target $p(W|\mathbf{I})$ is monotonic in terms of the Kullback-Leibler divergence.

Our Markov chain consists of μ pairs of reversible jumps denoted by

$$\mathbf{J}_m = (\mathbf{J}_{mr}, \mathbf{J}_{ml}),\ m = 1, 2, ..., \mu,$$

where $\mathbf{J}_{mr}$ and $\mathbf{J}_{ml}$ are the right and left jumps respectively. These reversible jumps implement operators on the curves and regions, such as death-birth, split-merge, switching models, switching partial relation order, grouping-ungrouping etc. Intuitively, a pair of reversible jumps is like a move that can be back-traced in AI search algorithms.

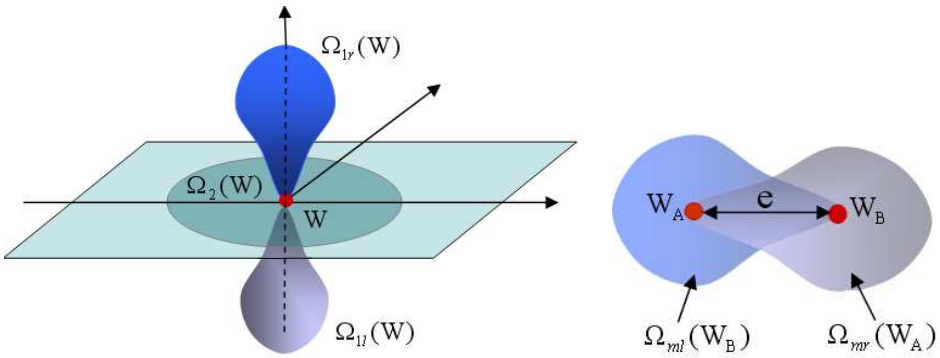
A pair of jumps $\mathbf{J}_m$ form a sub-kernel $\mathbf{K}_m$ which is a weighted sum of the right and left sub-kernels.

$$\mathbf{K}_m(W_A, W_B) = \omega_{mr}\mathbf{K}_{mr}(W_A, W_B) + \omega_{ml}\mathbf{K}_{ml}(W_A, W_B). \tag{19}$$

The overall kernel $\mathbf{K}$ is a linear summation of the sub-kernels

$$\mathbf{K}(W_A, W_B) = \sum_{m=1}^{\mu} \omega_m \mathbf{K}_m(W_A, W_B), \quad \omega_1 + \cdots + \omega_\mu = 1, \tag{20}$$

where $\omega_m, m = 1, 2, ..., \mu$ are the probability for choosing a specific move, and are time dependent. For example, we should use the birth operators more often at the beginning.



(a) (b)

Figure 5: (a). Illustrations for the scope $\Omega(W)$ at a state W. The round domain on the horizontal plane represents the scope of a pair of *symmetric jumps* and the dumbbells represent the left and right scopes of a pair of *asymmetric jumps*. (b). A reversible jump $\mathbf{J}_m$ between two states W_A and W_B. The overlap between $\Omega_{mr}(W_A)$ and $\Omega_{ml}(W_B)$ affects the speed.

3.1.2 The scopes of reversible jumps

Each jump step can only change 1-2 variables in W and thus most entries in the transition matrices $\mathbf{K}_m(W_A, W_B)$ are zero. We define the *scopes* of a jump in the following.

Definition 1 *At state $W \in \Omega$, the scopes of the right and left jumps are the sets of states connected to W by $\mathbf{J}_{mr}$ and $\mathbf{J}_{ml}$ respectively minus W itself,*

$$\begin{aligned} \Omega_{mr}(W) &= \{W' : \mathbf{K}_{mr}(W, W') > 0,\ W' \in \Omega,\ W' \neq W\}, \\ \Omega_{ml}(W) &= \{W' : \mathbf{K}_{ml}(W, W') > 0,\ W' \in \Omega,\ W' \neq W\}. \end{aligned}$$

The scope of $\mathbf{J}_m$ at W is $\Omega_m(W) = \Omega_{mr}(W) \cup \Omega_{ml}(W)$.

Thus with μ pairs of jumps each state W is connected to a set

$$\Omega(W) = \cup_{m=1}^{\mu} \Omega_m(W).$$

Fig. 5.(a) illustrates the scope $\Omega(W)$ with $\mu = 2$ jumps in a 3-dimensional space. The scope $\Omega(W)$ is often small in comparison to the entire state space Ω, due to limited number of available operators and the locality of the Markov chain. By analogy, one may imagine that the Markov chain walks in a dark space Ω with a flashlight and the landscape is defined by probability $p(W|\mathbf{I})$. Then $\Omega(W)$ is the scope of the flashlight at state W.

For the jumps being reversible, we have the following observation,

$$W_B \in \Omega_{mr}(W_A) \quad \text{if and only if} \quad W_A \in \Omega_{ml}(W_B), \quad \forall m. \tag{21}$$

Fig. 5.(b) shows the scopes of $\Omega_{ml}(W_B)$ and $\Omega_{mr}(W_A)$.

We classify a reversible jump as *symmetric* and *asymmetric* in the following. The design of the jumps will be affected by this property as we show in the subsection below.

1. A pair of reversible jumps are said to be symmetric if $\Omega_{mr}(W) = \Omega_{ml}(W)$ for $W \in \Omega$. An example is the jumps for switching image models of a region in next section where the scope is the image model space (family with parameters) and it is illustrated in Fig. 5.(a) by the round domain in the horizontal plane. Thus for any two connected states x and y, we have

$$W_B \in \Omega_{mr}(W_A) \cup \{W_A\} = \Omega_{ml}(W_B) \cup \{W_B\} \ni W_A. \tag{22}$$

2. A pair of reversible jumps are said to be asymmetric if $\Omega_{mr}(W) \cap \Omega_{ml}(W) = \emptyset$. For example, the death-birth, split-merge jumps have disjoint right and left scopes. In Fig. 5.(a) we illustrated $\Omega_{mr}(W)$ and $\Omega_{ml}(W)$ by the two dumb-bells respectively. In this case, for an edge $e = (W_A, W_B)$ shown in Fig. 5.(b) we have

$$W_B \in \Omega_{mr}(W_A) \cup \{W_A\} \neq \Omega_{ml}(W_B) \cup \{W_B\} \ni W_A. \tag{23}$$

   But the two sets $\Omega_{mr}(W_A)$ and $\Omega_{ml}(W_B)$ overlap, as Fig. 5.(b) displays. The overlap affects the Markov chain speed.

3.1.3 Gibbs and Metropolis-Hastings samplers

For a pair of symmetric jumps $\mathbf{J}_m$, we use the Gibbs sampler [15] to design $\mathbf{K}_m$ which will observe the invariance eqn.(16). At state W_A, we sample the next state W_A proportional to $p(W|\mathbf{I})$ in the scope $\Omega_{mr}(W) = \Omega_m(W)$, and the probability is normalized by the total probability in $\Omega_m(W)$ plus $p(W_A)$,

$$W_B \sim p_{mr,A}(W_B) = \begin{cases} \frac{p(W_B|\mathbf{I})}{p(W_A|\mathbf{I}) + \sum_{W_C \in \Omega_{mr}(W_A)} p(W_C|\mathbf{I})} & W_B \in \Omega_{mr}(W_A) \cup \{W_A\}, \\ 0 & W_B \notin \Omega_{mr}(W_A) \cup \{W_A\} \end{cases} \tag{24}$$

similarly it samples $W_A \sim p_{ml,B}(W_A)$ in the scope $\Omega_{ml}(W_y) = \Omega_m(W)$. Because $\mathbf{J}_m$ is symmetric (see eqn.22), the normalization (denominator in eqn.(24) is the same for both $p_{mr,A}(W_B)$ and $p_{ml,B}(W_A)$. They are canceled when they are plugged in the invariance equations (16).

This condition is unfortunately not observed for the asymmetric jumps. Therefore we design the asymmetric jumps according to the stronger condition – the detailed balance equations in (17). A standard way to satisfy the detailed balance equations is the Metropolis-Hastings design[29, 21].

$$\mathbf{K}_m(W_A, W_B) = \mathbf{Q}_{mr}(W_A, W_B)\alpha_{mr}(W_A, W_B), \quad \text{for } W_A \neq W_B,\ m = 1, 2, ..., \mu. \tag{25}$$

$\mathbf{Q}_{mr}(W_A, W_B) = \mathbf{Q}_{mr}(W_B|W_A)$ is a proposal (conditional) probability for moving from W_A to W_B with jump $\mathbf{J}_{mr}$ and $\alpha(W_A, W_B)$ is the acceptance probability,

$$\alpha_{mr}(W_A, W_B) = \min(1, \frac{\mathbf{Q}_{ml}(W_A|W_B)}{\mathbf{Q}_{mr}(W_B|W_A)} \cdot \frac{p(W_B|\mathbf{I})}{p(W_A|\mathbf{I})}). \tag{26}$$

It uses the target probability ratio $\frac{p(W_B|\mathbf{I})}{p(W_A|\mathbf{I})}$ to rectify the proposal probability ratio $\frac{\mathbf{Q}_{ml}(W_A|W_B)}{\mathbf{Q}_{mr}(W_B|W_A)}$. When the proposal is rejected, the Markov chain stays at state W_A. Thus

$$\mathbf{K}_{mr}(W_A, W_A) = 1 - \sum_{W_B \neq W_A} \mathbf{K}_{mr}(W_A, W_B), \quad \forall W_A \in \Omega. \tag{27}$$

It is easy to check that the detailed balance equation (17) follows from eqns.(25), (26) and (27). The key issue is to design the proposal probabilities Q_{mr}, Q_{ml} for fast computation. This the subject of the next two subsections.

3.2 The Metropolized Gibbs sampler

In this subsection, we study a design scheme called Metropolized Gibbs sampler which combines Metropolis-Hastings and Gibbs samplers. Basically it proposes a state W_B at W_A by a Gibbs sampler strategy within the scope $\mathbf{J}_{mr}(W_A)$ and then accepts the move by a Metropolis-Hastings step.

Consider the a pair of reversible jumps $\mathbf{J}_m = (\mathbf{J}_{mr}, \mathbf{J}_{ml})$ between two states W_A and W_B, we design a pair of proposal probabilities following the target probability p normalized within the scopes.

$$\mathbf{Q}^*_{mr}(W_B|W_A) = \frac{p(W_B|\mathbf{I})}{\sum_{W_C \in \Omega_{mr}(W_A)} p(W_C|\mathbf{I})}, \quad \text{for } W_B \in \Omega_{mr}(W_A) \tag{28}$$

$$\mathbf{Q}^*_{ml}(W_A|W_B) = \frac{p(W_A|\mathbf{I})}{\sum_{W_C \in \Omega_{ml}(W_B)} p(W_C|\mathbf{I})}, \quad \text{for } W_A \in \Omega_{ml}(W_B). \tag{29}$$

We set $\mathbf{Q}^*_{mr}(W_B|W_A) = 0$ and $\mathbf{Q}^*_{ml}(W_A|W_B) = 0$ outside the two scopes respectively. Following eqns.(25) and (26), we have the following transition kernel,

$$\begin{aligned}
\mathbf{K}_{mr}(W_A, W_B) &= p(W_B|\mathbf{I}) \min(\frac{\mathbf{Q}^*_{mr}(W_B|W_A)}{p(W_B|\mathbf{I})}, \frac{\mathbf{Q}^*_{ml}(W_A|W_B)}{p(W_A|\mathbf{I})}), \\
&= p(W_B|\mathbf{I}) \min(\frac{1}{\sum\limits_{W_C \in \Omega_{mr}(W_A)} p(W_C|\mathbf{I})}, \frac{1}{\sum\limits_{W_C \in \Omega_{ml}(W_B)} p(W_C|\mathbf{I})})
\end{aligned}$$

The proposal probability is the same as the Gibbs sampler in eqn.(24), except that we set $\mathbf{Q}^*_{\mathrm{m}}(W_A, W_A) = 0, \ \forall W_A$ (note that $\mathbf{K}(W_A, W_A) \neq 0$) and the normalization factor is thus changed accordingly. Ideally if the scope is large, the probabilities in the denominators sum to one, and $\mathbf{K}_{mr}(W_A, W_B)$ is close to $p(W_B|\mathbf{I})$. Thus it generates fair samples rapidly.

This design is called the *Metropolized Gibbs sampler* (MGS) following a simple example in [25], because it uses a Metropolis-Hasting step to rectify the proposal $\mathbf{Q}^*$ designed by a Gibbs sampler over the scope.

3.3 Approximating the MGS proposal by discriminative models

The computational time is decided by two factors: (1) the mixing time or first hitting time measured by the number of steps t; (2). the computational cost at each step. The former demands large jumps scopes and good proposal probabilities, and the latter requires fast computation of the proposal probabilities.

In eqns.(28) and (29), the MGS proposals $\mathbf{Q}^*_{mr}(W_x|W_y)$ and $\mathbf{Q}^*_{ml}(W_y|W_x)$ compute the target probability $p(W_C|\mathbf{I})$ over two scopes $W_C \in \Omega_{mr}(W_A) \cup \Omega_{ml}(W_B)$. We observe that

$$\begin{aligned}
\mathbf{Q}^*_{mr}(W_B|W_A) &= \frac{p(W_B|\mathbf{I})}{\sum\limits_{W_C \in \Omega_{mr}(W_A)} p(W_C|\mathbf{I})} = \frac{\frac{p(W_B|\mathbf{I})}{p(W_A|\mathbf{I})}}{\sum\limits_{W_C \in \Omega_{mr}(W_A)} \frac{p(W_C|\mathbf{I})}{p(W_A|\mathbf{I})}}, \quad \text{for } W_B \in \Omega_{mr}(W_A) \\
\mathbf{Q}^*_{ml}(W_A|W_B) &= \frac{p(W_A|\mathbf{I})}{\sum\limits_{W_C \in \Omega_{ml}(W_B)} p(W_C|\mathbf{I})} = \frac{\frac{p(W_A|\mathbf{I})}{p(W_B|\mathbf{I})}}{\sum\limits_{W_C \in \Omega_{ml}(W_B)} \frac{p(W_C|\mathbf{I})}{p(W_B|\mathbf{I})}}, \quad \text{for } W_A \in \Omega_{ml}(W_B).
\end{aligned}$$

While it is hard to compute $p(W_C|\mathbf{I}))$ for every state $W_C \in \Omega_{mr}(W_A) \cup \Omega_{ml}(W_B)$ at each step since W_C contains many variables, it is much easier to compute the ratio $\frac{p(W_C)}{p(W_A)}$ or $\frac{p(W_B)}{p(W_A)}$. Because the states W_A and W_C differ in just 1-2 items locally, most of the terms are thus canceled in the ratio.

Thus we approximate the MGS proposals in two steps so that they can be computed effectively. Note that this approximation only changes the design of the proposal probabilities, then the new proposal probabilities have to be accepted by the Metropolis-Hastings step. Thus the detailed balance equations are still observed. The kernel $\mathbf{K}_m, m = 1, 2, ..., \mu$ still have the invariant probability $p(W|\mathbf{I})$.

Firstly, the posterior probability ratios $\frac{p(W_A)}{p(W_B)}$ can be written in factorized form and we approximate each factor by discriminative posterior probability ratios. The discriminative probability ratio is computed by bottom-up methods and is treated as the weight of each candidate.

$$\omega_{mr,A}(W_C) \approx \frac{p(W_C|\mathbf{I})}{p(W_A|\mathbf{I})}, \text{ for } W_C \in \Omega_{mr}(W_A), \ \omega_{ml,B}(W_C) \approx \frac{p(W_C|\mathbf{I})}{p(W_B|\mathbf{I})}, \text{ for } W_C \in \Omega_{ml}(W_B).$$

Secondly, we replace the two continuous scopes $\Omega_{mr}(W_A)$ and $\Omega_{ml}(W_B)$ by two finite sets of "particles" $S_{mr}(W_A)$ and $S_{ml}(W_B)$ respectively. A particle is a candidate with non-trivial weight. As Fig.12 illustrates, these particles represent the promising candidates in the scopes.

Therefore the new proposal probabilities become,

$$\begin{aligned}
\mathbf{Q}_{mr}(W_B|W_A) &= \frac{\omega_{mr,A}(W_B)}{\sum_{W_C \in S_{mr}(W_A)} \omega_{mr,A}(W_C)}, \quad \text{for } W_B \in \Omega_{ml}(W_A). \\
\mathbf{Q}_{ml}(W_A|W_B) &= \frac{\omega_{ml,B}(W_A)}{\sum_{W_C \in S_{ml}(W_B)} \omega_{ml,B}(W_C)}, \quad \text{for } W_A \in \Omega_{ml}(W_B).
\end{aligned}$$

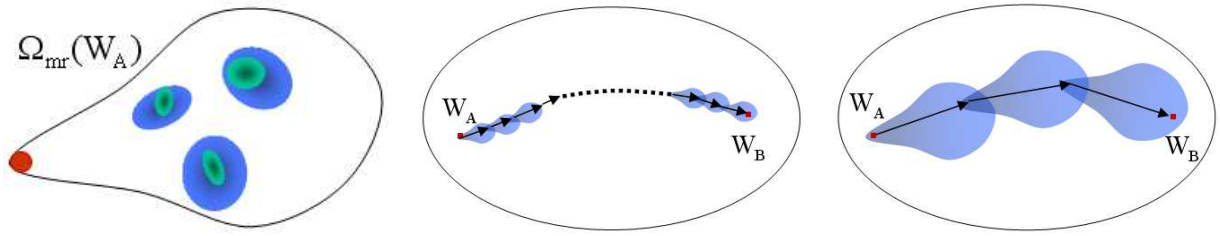
The weight $\omega_{mr,A}(W_B)$ for a candidate state $W_B \in \Omega_{mr}(W_A)$ depends on the current state W_A. As we shall show in the next section, each pair of reversible jumps maintains a set of candidates whose weights are updated on-line in the computational process.

The transition kernel for jump $\mathbf{J}_{mr}$ from W_A to W_B is then

$$\mathbf{K}_{mr}(W_A, W_B) = \mathbf{Q}_{mr}(W_B|W_A) \min(1, \frac{\mathbf{Q}_{ml}(W_A|W_B)}{\mathbf{Q}_{mr}(W_B|W_A)} \cdot \frac{p(W_B|\mathbf{I})}{p(W_A|\mathbf{I})}). \tag{30}$$

In computer vision and machine learning, there are abundant discriminative methods that can compute the weights in various subspaces to approximate the posterior ratios. For example, it was proved that the popular boosting methods for classification converge to the posterior probability ratio on the class labels [32] as the number of features increases.

Figure 6.(a) shows the approximation of the true posterior ratio by weights ω in a scope $\Omega_{mr}(W_A)$.



(a) (b) (c)

Figure 6: (a) Illustration of posterior probability p in scope $\Omega_{mr}(W_A)$ covered by the proposal probability $\mathbf{Q}_{mr}$. Darker points have high probabilities. (b). With small scopes of jumps there will be a long path or more steps between some states W_A and W_B. (b). The composite jumps enlarge the scope of each jump and empirically result in shorter paths and less steps.

With the ease of computing proposal probabilities, we seek to enlarge the scope of some jumps so that the Markov chain mix rapidly. This idea is illustrated in Figures 6.(b) and 6.(c). With small scopes, the Markov chain needs more steps from a state W_A to a state and W_B. If W_A and W_B are two distinct modes (maxima), this path will have very small probability to occur. With enlarged scopes, at each step, the proposal is generated over a long distance and thus the Markov chain may move between state W_A and W_B in less steps and the probability for jumping between two distinct modes increases. By analogy, when the jump scopes are small, it is like searching in a dark space with a flashlight, and when the scope is large, it is like searching with a long range Radar.

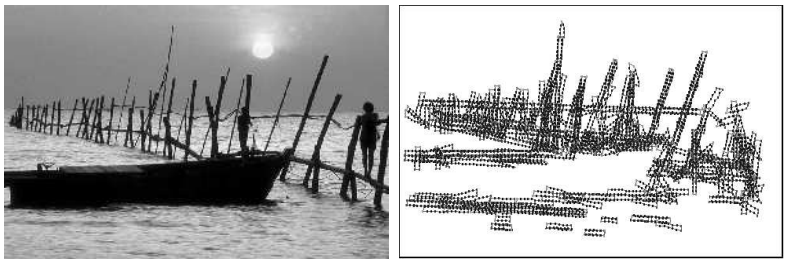
In the next section, we use a Swendsen-Wang cut (SWC) method [2] to design composite jumps where the proposal is a product of many simple proposals. Such jumps realize split-merge of large regions, grouping and ungrouping a set of curves into parallel groups and trees, as they can move a set of elements at a single step.

4 Designing jumps using approximated MGS

In this section, we study seven pairs of reversible jumps using the approximated MGS design discussed in the previous section for curve detection and grouping: (1) death-birth of an atomic curve, (2) split-merge of a simple curve, (3) switch partial order between two simple curves, (4) switching between a degenerated region and a curve, (5) switching intensity models, (6) grouping/ungrouping parallel curves, (7) grouping/ungrouping tree. (1-5) are simple jumps and (6-7)

are composite jumps. The reversible jumps for regions are referred to a previous paper [37].

4.1 Bottom-up computation of the candidate set and weights for simple jumps on curves



(a) input image (b) curve candidates

Figure 7: (a) An example image. (b) Atomic curves (oriented bars) computed by a matching pursuit detection and they are used as proposal candidates.

The key idea in the DDMCMC framework is the use of discriminative models to compute the proposal probabilities, and some discussions about the interaction between discriminative and generative models are referred to [39]. This paper is mostly focuses on the curve part. We use the matching pursuit algorithm [28] to detect bars in a bottom-up step. An example is shown in Figure (7).

In the discrete form, a curve C consists of a number of oriented bars in a chain (see Fig. 3.(b)). In a bottom-up process we compute an excessive number of candidates by a matching pursuit algorithm [28]. The matching pursuit algorithm convolves the image with an oriented bar at each location (x_i, y_i) and a number of discretized angles θ_i, and the bars have certain width w_i and constant intensity f_i. Large responses mean high likelihood that a curve passing through the location with tangent directions coinciding the bar orientation. By setting a sufficiently low threshold, we obtain a set of "atomic curves" as bottom-up candidates in the jump type I discussed in Section (4.2).

$$\Delta_c^{\mathrm{DD}} = \{c_i = (x_i, y_i, \theta_i, w_i, f_i) : i = 1, 2, ..., M_c^{\mathrm{DD}}\}.$$

where (x_i, y_i, θ_i) is the center and orientation, w_i the width and f_i the intensity. Each c_i has a weight ω_i which measures the fitness of c_i in the domain $\mathcal{D}(c_i)$. $\omega_i = p(\mathbf{I}_{\mathcal{D}(c_i)}|c_i)$

Thus we have a set of weighted atomic curves.

$$S_c = \{(c_i, \omega_i) : \ i = 1, 2, ..., M_c^{\mathrm{DD}}\}. \tag{31}$$

An example is shown in Fig. 7.b where each atomic curve is represented by a bar. A new curve will be created by selecting an atomic curve in the set S_c or an existing curve can be extended by attaching an atomic curve to one of its ends.

The detection of atomic curves can be reformulated as computing discriminative models $p(\ell|F(\mathbf{I}))$ where $\ell \in \{+1, -1\}$ is the label for "curve" or "non-curve" respectively and $F(\mathbf{I})$ denotes a number of features in detection. By setting a low threshold in the ratio test $\frac{p(\ell=+1|F(\mathbf{I})}{p(\ell=-1|F(\mathbf{I})}$ we can put all non-trivial candidates as particles in the set.

4.2 Jump pair I: death and birth of an atomic curve

$\mathbf{J}_1 = (\mathbf{J}_{1r}, \mathbf{J}_{1l})$ is a pair of jumps for adding an atomic curve from S_c or removing one from the existing curves in current W. Adding an atomic curve results into two possible situations: (1) The added atomic curve becomes a new curve itself or (2) it can be attached to one end of an existing curve, if it is near one. Likewise, an existing atomic curve can be removed if it is either on a curve with no other atomic curve or it is on one of the two ends of a curve. This simulates a birth-death process between two states W_A and W_B,

$$W_A = (W_-, K^c_A, \mathcal{PR}_A) \;\; \rightleftharpoons \;\; (W_-, c_B, K^c_B(c_B), \mathcal{PR}_B(c_B)) = W_B.$$

In the above notation, c_B is the new-born atomic curve, W_- denotes all other variables not changed in this reversible jump and they are the same for both W_A and W_B. The total number of curves K^c and the partial order relation $\mathcal{PR}$ may change depending on whether c_B is an independent curve or merely an extension of an existing curve.

Fig. 8 shows an example. At state W_A the birth jump has 8 candidate atomic curves (shown by ellipses in the upper middle) and one is proposed as c_B in W_B. Conversely, at state W_B the death jump has 5 candidates (in lower middle) and c_B is proposed. The birth and death jumps have different scopes $\Omega_{1r}(W_A) \neq \Omega_{1l}(W_B)$, and thus are asymmetric,

To design the reversible jump, we calculate the proposal probabilities following the Metropolized Gibbs sampler (MGS) design and then approximate them by discriminative models in a factorized form.

We first consider the birth jump. For any state $W \in \Omega_{1r}(W_A)$, it has an extra atomic curve c_{1r} and we denote $W = (W_-, c_{1r}, K^c_{1r}, \mathcal{PR}_+)$. $W = W_B$ is an instance in $\Omega_{1r}(W_A)$ when $c_{1r} = c_B$. The MGS proposal probability for selecting c_B is a conditional posterior probability over the jump

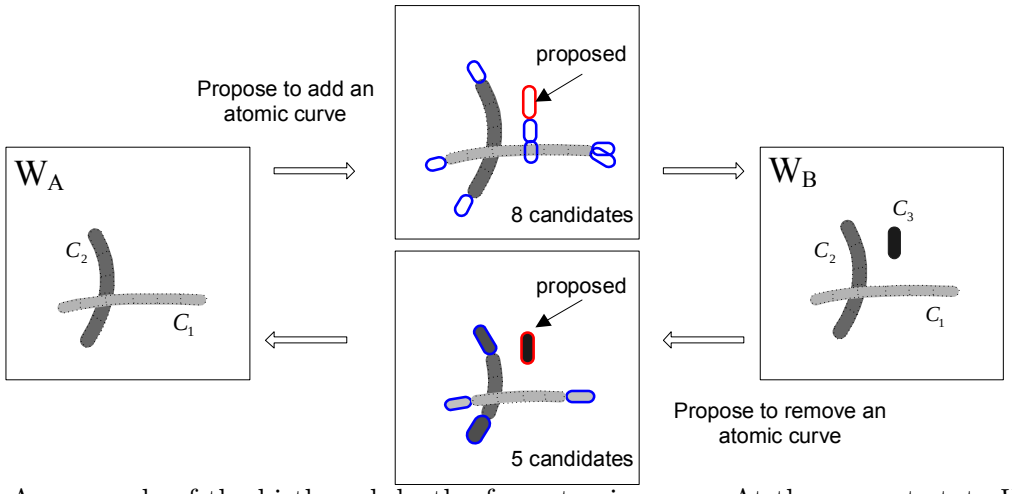


Figure 8: An example of the birth and death of an atomic curve. At the current state W_A, there are 8 possible atomic curves to be proposed, which are shown as ellipses in the upper middle figure. W_B is the state after selecting an atomic curve and make it as a new curve. From W_B to W_A, there are 5 candidate atomic curves to be removed, which are shown in the lower middle figure. Choosing the same atomic curve changes W_B back to W_A.

scope $\Omega_{mr}(W_A)$,

$$\mathbf{Q}^*_{1r}(W_B|W_A) = \frac{p(W_B|\mathbf{I})}{\sum_{W\in\Omega_{1r}(W_A)} p(W|\mathbf{I})} = \frac{\frac{p(W_B|\mathbf{I})}{p(W_A|\mathbf{I})}}{\sum_{c_{1r}\in S_{1r}} \frac{p(W|\mathbf{I})}{p(W_A|\mathbf{I})}} \tag{32}$$

We divide both the nominator and the denominator by $p(W_A|\mathbf{I})$ for the probability ratios are much easier to compute due to cancellation. Observe that the likelihoods $p(\mathbf{I}|W)$ and $p(\mathbf{I}|W_A)$ differ only in the way they explain pixels covered by c_{1r} in a domain $\mathcal{D}(c_{1r})$. The former explains $\mathbf{I}_{\mathcal{D}(c_{1r})}$ by the new model in c_{1r} while the latter explains $\mathbf{I}_{\mathcal{D}(c_{1r})}$ by some region $R(c_{1r})$ which depends on c_{1r}. Therefore

$$\frac{p(\mathbf{I}|W)}{p(\mathbf{I}|W_A)} = \frac{p(\mathbf{I}_{\mathcal{D}(c_{1r})}|c_{1r})}{p(\mathbf{I}_{\mathcal{D}(c_{1r})}|R(c_{1r}))}.$$

We can rewrite the posterior probability ratios in a factorized form,

$$\frac{p(W|\mathbf{I})}{p(W_A|\mathbf{I})} = \frac{p(\mathbf{I}|W)p(W)}{p(\mathbf{I}|W_A)p(W_A)} = \frac{p(\mathbf{I}_{\mathcal{D}(c_{1r})}|c_{1r})}{p(\mathbf{I}_{\mathcal{D}(c_{1r})}|R(c_{1r}))} \cdot p(c_{1r}|W_-) \cdot \frac{p(K^c(c_{1r})|c_{1r})}{p(K^c_A)} \cdot \frac{p(\mathcal{PR}(c_{1r})|c_{1r})}{p(\mathcal{PR}_A)}. \tag{33}$$

We are only interested in atomic curves which have non-trivial probability ratios. Two types of atomic curves have non-trivial probability ratios: (1). elements detected in the bottom-up step which have non-trivial likelihood ratio. (2). elements suggested by context based on continuity of existing curves. For example, Fig.8 has 8 candidates in $\hat{\Delta}_{1r}(W_A)$, five of which extend existing curves and thus are proposed by the context.

For each candidate atomic curve $c_{1r}^{(i)}$, its weight $\hat{\omega_{1r}}^{(i)}$ approximates the factorized ratio in eqn.(33). Intuitively, the weight of a candidate c_{1r} is a product of three factors. (1). How well the data is fitted by the current model, $p(\mathbf{I}_{\mathcal{D}(c_{1r})}|R(c_{1r}))$. This available for the current W_A since we compute each term in eqn. (15) for each existing region and curve. (2). Its fitness to data $\mathbf{I}_{\mathcal{D}(c_{1r})}$ which is either computed for the data-driven candidates or from the context, $p(\mathbf{I}_{\mathcal{D}(c_{1r})}|c_{1r})$. This is approximated based on how good a local Gabor function fits the image, same as in the matching pursuit algorithm. (3). The possible change of curve number and partial order relation, $\frac{p(\mathcal{PR}(c_{1r})|c_{1r})}{p(\mathcal{PR}_A)}$. This is approximated by a uniform distribution. Thus, we have a set of weighted candidates for birth at W_A.

$$S_{1r}(W_A) = \{\ (c_{1r}^{(i)}, \omega_{1r}^{(i)}) :\ i = 1, 2, ..., N_{1r}.\}$$

$(c_{1r}^{(B)}, \omega_{1r}^{(B)})$ is one instance in the above set that leads to state W_B. Then the proposal probability is

$$\mathbf{Q}_{1r}(W_B|W_A) = \frac{\omega_{1r}^{(B)}}{\sum_{i=1}^{N_{1r}} \omega_{1r}^{(i)}}.$$

In a similar way, we can design the death jump $\mathbf{J}_{1l}$ from W_B to W_A. Let $\Omega_{1l}(W_B)$ be the jump scope, any $W \in \Omega_{1l}(W_B)$ will have one atomic curve c_{1l} removed from W_B. We are interested in computing the probability for proposing W_A from $\Omega_{1l}(W_B)$. According to the MGS design, it is

$$\mathbf{Q}_{1l}^*(W_A|W_B) = \frac{p(W_A|\mathbf{I})}{\sum_{W\in\Omega_{1l}(W_B)} p(W|\mathbf{I})} = \frac{\frac{p(W_A|\mathbf{I})}{p(W_B|\mathbf{I})}}{\sum_{c_{1l}\in S_{1l}} \frac{p(W|\mathbf{I})}{p(W_B|\mathbf{I})}} \tag{34}$$

The likelihoods $p(\mathbf{I}|W)$ and $p(\mathbf{I}|W_B)$ differ only in the way they explain pixels covered by c_{1l} in a domain $\mathcal{D}(c_{1l})$. Therefore we have

$$\frac{p(\mathbf{I}|W)}{p(\mathbf{I}|W_B)} = \frac{p(\mathbf{I}_{\mathcal{D}(c_{1l})}|R(c_{1l})}{p(\mathbf{I}_{\mathcal{D}(c_{1l})}|R(c_{1l}))},$$

where $R(c_{1l})$ is the region explaining $\mathbf{I}_{\mathcal{D}(c_{1l})}$ in W. Thus the posterior probability ratios can be rewritten in a factorized form,

$$\frac{p(W|\mathbf{I})}{p(W_B|\mathbf{I})} = \frac{p(\mathbf{I}|W)p(W)}{p(\mathbf{I}|W_B)p(W_B)} = \frac{p(\mathbf{I}_{\mathcal{D}(c_{1l})}|R(c_{1l}))}{p(\mathbf{I}_{\mathcal{D}(c_{1l})}|c_{1l})} \cdot \frac{1}{p(c_{1l}|W_-)} \cdot \frac{p(K^c(c_{1l})|c_{1l})}{p(K_B^c)} \cdot \frac{p(\mathcal{PR}(c_{1l})|c_{1l})}{p(\mathcal{PR}_B)}. \tag{35}$$

Unlike the birth jump, the candidate set S_{1l} contains only short atomic curves at the ends of the existing curves. For example, $|S_{1l}| = N_{1r} = 5$ in Fig.8. Thus we maintain a set of weighted candidates,

$$S_{1l}(W_B) = \{\ (c_{1l}^{(i)}, \omega_{1l}^{(i)}) :\ i = 1, 2, ..., N_{1l}.\}$$

The weight $\omega_{1l}^{(i)}$ is computed according to eqn.(35) where the factors have very intuitive meanings. $p(\mathbf{I}_{\mathcal{D}(c_{1l})}|R(c_{1l}))$ is computed using the image model of underlying region which c_{1l} occludes. Intuitively, it is to say an atomic curve whose image part can not be fitted very well by its occluded region model should have low chance to be removed. $p(\mathbf{I}_{\mathcal{D}(c_{1l})}|c_{1l})$ is available since it is computed in eqn. (15). $p(c_{1l}|W_-)$ is the prior of the atomic curve which is computed in eqn. (14).

$(c_{1l}^{(A)}, \omega_{1l}^{(A)})$ is one instance in the above set, when it is removed, then state W_B becomes W_A. Then the proposal probability is

$$\mathbf{Q}_{1l}(W_A|W_B) = \frac{\omega_{1l}^{(A)}}{\sum_{i=1}^{N_{1l}} \omega_{1l}^{(i)}}.$$

Finally the birth and death proposal probabilities are corrected in a Metropolis-Hastings step.

$$\begin{aligned}
\mathbf{K}_{1r}(W_A, W_B) &= Q_{1r}(W_B|W_A)\min(1, \frac{Q_{1l}(W_A|W_B)}{Q_{1r}(W_B|W_A)} \cdot \frac{p(W_B|\mathbf{I})}{p(W_A|\mathbf{I})}), \\
\mathbf{K}_{1l}(W_B, W_A) &= Q_{1l}(W_A|W_B)\min(1, \frac{Q_{1r}(W_B|W_A)}{Q_{1l}(W_A|W_B)} \cdot \frac{p(W_A|\mathbf{I})}{p(W_B|\mathbf{I})}).
\end{aligned}$$

4.3 Jump II: split and merge of curves

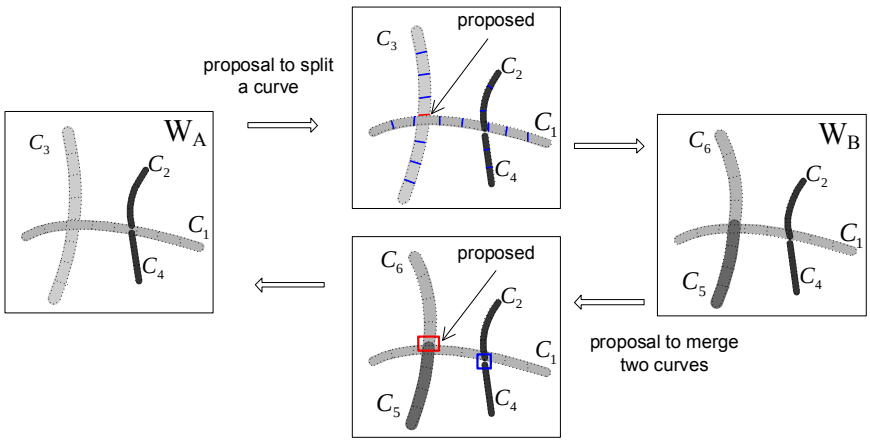


Figure 9: An example of the split-merge jumps for simple curves. At state W_A, a set of 18 candidate sites are shown in the upper-middle figure for splitting the simple curves and one is proposed probabilistically. Splitting C_3 into two curves C_5 and C_6, we obtain state W_B. At W_B, a set of 2 candidate pairs shown in the lower-middle figure are listed for merge.

The second pair of reversible jumps $\mathbf{J}_2 = (\mathbf{J}_{2r}, \mathbf{J}_{2l})$ realizes split-merge for simple curves. Fig. 9 shows an example of $\mathbf{J}_2$ between two states W_A and W_B. In a way similar to the birth-death jumps, we maintain two candidates lists at the current state to approximate the jump scopes $\Omega_{2r}(W_A)$ and

$\Omega_{2l}(W_B)$ respectively.

$$S_{2r}(W_A) = \{\ (z_{2r}^{(i)}, \omega_{2r}^{(i)}) :\ i = 1, 2, ..., N_{2r}.\}, \quad S_{2l}(W_B) = \{\ (z_{2l}^{(i)}, \omega_{2l}^{(i)}) :\ i = 1, 2, ..., N_{2l}.\}$$

We adopt a discrete notion with $z_{2r}^{(i)}$ and $z_{2l}^{(i)}$ being the site between adjacent atomic curves for split and merge respectively. For example, Fig. 9 shows 18 candidate sites for split at W_A and 2 sites for merge at W_B.

The MGS proposal for split is,

$$\mathbf{Q}_{2r}^{*}(W_B|W_A) = \frac{p(W_B|\mathbf{I})}{\sum_{W\in\Omega_{2r}(W_A)} p(W|\mathbf{I})} = \frac{\frac{p(W_B|\mathbf{I})}{p(W_A|\mathbf{I})}}{\sum_{x_s\in S_{2r}(W_A)} \frac{p(W|\mathbf{I})}{p(W_A|\mathbf{I})}}.$$

Each site $z_{mr} \in S_{2r}(W_A)$ corresponds to a state $W \in \Omega_{2r}(W_A)$. W differs from W_A by splitting a curve C_k into two curves C_i and C_j. Accordingly the number of free curves increase by 1 and the partial order may be changed.

$$W_A = (K^c,\ C_k, \mathcal{PR}, W_-) \longrightarrow W = (K^c + 1,\ C_i,\ C_j,\ \mathcal{PR}', W_-).$$

We write the posterior ratio in a factorized form, noting that these free curves are independent of other variables in W_-.

$$\frac{p(W|\mathbf{I})}{p(W_A|\mathbf{I})} = \frac{p(\mathbf{I}_{\mathcal{D}(C_i)\cup\mathcal{D}(C_j)})|C_i, C_j)}{p(\mathbf{I}_{\mathcal{D}(C_k)}|C_k)} \cdot \frac{p(C_i)p(C_j)}{p(C_k)} \cdot \frac{p(K^c+1)}{p(K^c)} \cdot \frac{p(\mathcal{PR}'|C_i, C_j)}{p(\mathcal{PR}|C_K)}$$

The four factors, again, have very intuitive meanings. (1). $\frac{p(\mathbf{I}_{\mathcal{D}(C_i)\cup\mathcal{D}(C_j)})|C_i,C_j)}{p(\mathbf{I}_{\mathcal{D}(C_k)}|C_k)}$ measures the goodness of fit for the curves before and after splitting. It decides, probabilistically, which curve to split and where to make the split. Intuitively, if the intensity model of a curve C_k does not fit the image very well, then C_k should have high probability to be split. This is directly available in $p(\mathbf{I}_{\mathcal{D}(C_k)}|C_k)$ for the current W_A. For curve C_k, there are many places to make the split, depending upon on how many atomic curves it has. If its two possible segments $\mathbf{I}_{\mathcal{D}(C_i)}$ and $\mathbf{I}_{\mathcal{D}(C_j)}$ are very different in appearances, their connection site should have more chance to be proposed. This is represented by $p(\mathbf{I}_{\mathcal{D}(C_i)\cup\mathcal{D}(C_j)})|C_i, C_j)$ which is approximated by a Gaussian kernel, $G(D(\mathbf{I}_{\mathcal{D}(C_i)}||\mathbf{I}_{\mathcal{D}(C_j)}), \sigma^2)$. $D(\mathbf{I}_{\mathcal{D}(C_i)}||\mathbf{I}_{\mathcal{D}(C_j)})$ is a distance measure for the similarity between the two segments, $\mathbf{I}_{\mathcal{D}(C_i)}$ and $\mathbf{I}_{\mathcal{D}(C_j)}$. For fast computation, we use mean of the image segments to measure the distance. (2). $\frac{p(C_i)p(C_j)}{p(C_k)}$ reflects the prior of curves only and does not involve with the image. Similar as in (1), $p(C_k)$ is directly available in computing $p(W_A)$. If a curve is not so smooth, then it should have high

probability to be proposed. $p(C_i)p(C_j)$ can be also computed when computing $p(C_k)$ in eqn. (3). Intuitively, the site which splits a curve into two smooth segments should have more chance to be proposed. (3). Priors on the curve number can be directly computed. (4). Priors on the partial order $p(\mathcal{PR}'|C_i, C_j)$ is approximated by a uniform distribution. The weight $\omega_{2r}^{(i)}$ will approximate the ratio $\frac{p(W|\mathbf{I})}{p(W_A|\mathbf{I})}$. The proposal probability is the weight normalized in the candidate set,

$$\mathbf{Q}_{2r}(W_B|W_A) = \frac{\omega_{2r}^{(B)}}{\sum_{i=1}^{N_{2r}} \omega_{2r}^{(i)}}.$$

$\omega_{2r}^{(B)}$ is the weight for a site $z_{2r}^{(B)} \in S_{2r}(W_A)$ that leads to state W_B.

Similarly we update the weights in the candidate set $S_{2l}(W_B)$ and compute the proposal probability $\mathbf{Q}_{2r}(W_A|W_B)$ as the normalized weight.

4.4 Jump III: swap the partial order relation

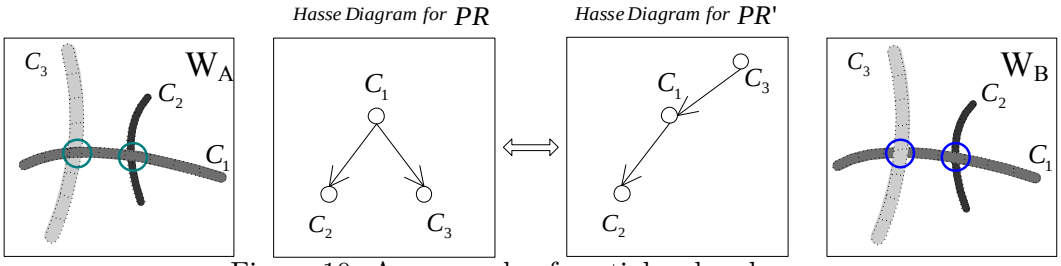


Figure 10: An example of partial order change.

The third jump $\mathbf{J}_3 = (\mathbf{J}_{3r}, \mathbf{J}_{3l})$ is a pair of symmetric jumps that switch partial order relation between curves. The candidate sets S_{3r} and S_{3l} are the same for the left and right moves.

$$S_3 = \{(z_3^{(k)}, \omega_3^{(k)}) : \quad z_3^{(i)} =< C_i, C_j > \in \mathcal{PR}, k = 1, 2, ..., N_3.\}$$

Each candidate $z_3^{(i)}$ is an occlusion between two existing curves $C_i \preceq C_j$ and the jump is to reverse the order

$$W_A = (< C_i, C_j >, W_-) \rightleftharpoons (< C_j, C_i >, W_-) = W_B,$$

Figure 10 shows an example of such a change of partial order. The weight of each candidate $z_3^{(i)}$ is only decided by the probability ratio on the overlapping image domain

$$\omega_3^{(k)} = \frac{p(W|\mathbf{I})}{p(W_A|\mathbf{I})} = \frac{p(\mathbf{I}_{\mathcal{D}(C_i)\cap\mathcal{D}(C_j)}|C_j)}{p(\mathbf{I}_{\mathcal{D}(C_i)\cap\mathcal{D}(C_j)}|C_j)}, \quad \forall k.$$

The position of the intersection points between C_i, C_j are collected in a candidate set Ξ of junctions. Each intersection has an equal probability in the candidate set S_3. Then we can compute the proposal probabilities for $\mathbf{J}_{3r}$ and $\mathbf{J}_{3l}$ in the same way as $\mathbf{J}_1$ and $\mathbf{J}_2$.

4.5 Jump IV: switching between degenerated region and curve

The fourth pair of jumps $\mathbf{J}_4 = (\mathbf{J}_{4r}, \mathbf{J}_{4l})$ is needed for resolving region-curve ambiguity. At a certain stage a region may become elongated and thin, thus it may switch to a curve. Conversely a short curve may become thick and switch to a region. The switch is realized by reversible jumps between two states,

$$W_A = (K^c - 1,\ K^r + 1, R_k, W_-) \ \rightleftharpoons \ (K^c,\ C_k, K^r, W_-) = W_B.$$

To do so, we maintain two weighted lists for degraded regions and curves respectively in the current state,

$$S_{4r}(W_A) = \{\, (R_{4r}^{(i)}, \omega_{4r}^{(i)}) : i = 1, 2, ..., N_{4r} \,\}, \quad S_{4l}(W_A) = \{\, (C_{4l}^{(i)}, \omega_{4l}^{(i)}) : i = 1, 2, ..., N_{4l} \,\}.$$

The weights are decided by the priors for curve and region.

$$\omega_{4r}^{(i)} = \frac{p(C_k)}{p(R_k)} \cdot \frac{p(K^c)}{p(K^c - 1)} \cdot p(K^r)p(K^r + 1), \quad \omega_{4l}^{(i)} = \frac{p(R_k)}{p(C_k)} \cdot \frac{p(K^c - 1)}{p(K^c)} \cdot p(K^r + 1)p(K^r),$$

In the same way, the proposal probabilities are computed as the normalized weights within the two candidate set. Since there is only prior about curves and regions involved, both $p(C_k)$ and $p(R_k)$ can be quickly computed at each step for all the regions and curves. Some simpler tests, such as measurement of aspect ratio and area, can be adopted to make the computation even faster. Intuitively, an elongated region will have high probability to be switched into a curve, and a "fat" curve will have a big chance to be turned into a region.

4.6 Jump V: switching intensity models

For each region or simple curve, we need to select a suitable generative model. For example, a region could be fitted to a texture, a color, a smooth shading model, or a clutter model in DDMCMC segmentation[37]. Each type of model has a parameter space which has multiple modes. So the fifth jump $\mathbf{J}_5 = (\mathbf{J}_{5r}, \mathbf{J}_{5l})$ realize the switch of models for each region or curve at a time. $\mathbf{J}_5$ is a symmetric jump with its scope being the parameter space of the models.

We compute a set of candidates by bottom-up methods, such as mean shift[8] in the parameter space of each type of models. Each candidate $(\ell^{(i)}, \theta^{(i)})$ is a mode in parameter space of type $\ell^{(i)}$.

$$\hat{S}_5 = \{(\ell^{(i)}, \theta^{(i)}, \omega_5^{(i)}) : \ i = 1, 2, ..., N_5.\}$$

When we switch models for a region R or a curve C, the weight $\omega^{(i)}$ for candidate model $(\ell^{(i)}, \theta^{(i)})$ is the accumulative votes from the pixels inside the domain $\mathcal{D}(R)$ or $\mathcal{D}(C)$. Each pixel contributes a vote in $[0, 1]$ depending on its goodness of fit to the candidate model. The vote is computed once in bottom-up. Then one model is proposed in proportion to its weight. Details about this part can be found in [37].

4.7 Jump VI: split and merge of trees

The sixth pair of jumps is the split-merge of tree structures, and thus it jumps between two states,

$$W_A = (W_-, T_A) \rightleftharpoons (W_-, T_B, T_C) = W_B$$

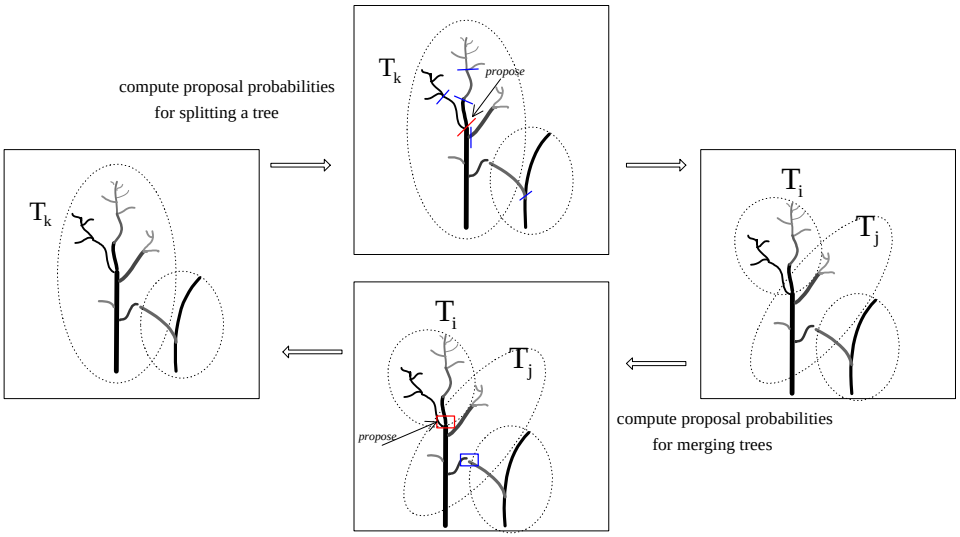


Figure 11: An example of splitting and merging trees.

Figure 11 illustrates an example where a tree T_k is split into trees T_i and T_j. Unlike the split-merge process for parallel curves, cutting the parent-child relation between any two curves in a tree will naturally split this tree into two. Thus, the process of splitting a tree has an analogy to that of splitting a curve. Similarly, we maintain two candidate lists at the current state to approximate the jump scopes $\Omega_{6r}(W_A)$ and $\Omega_{6l}(W_B)$ respectively.

$$S_{6r}(W_A) = \{ \ (z_{6r}^{(i)}, \omega_{6r}^{(i)}) : \ i = 1, 2, ..., N_{8r}.\}, \quad S_{6l}(W_B) = \{ \ (z_{6l}^{(i)}, \omega_{6l}^{(i)}) : \ i = 1, 2, ..., N_{8l}.\},$$

where $z_{6r}^{(i)}$ and $z_{6l}^{(i)}$ are sites between adjacent curves for tree split and merge respectively.

The MGS proposal for tree split is,

$$\mathbf{Q}_{6r}^{*}(W_B|W_A) = \frac{p(W_B|\mathbf{I})}{\sum_{W\in\Omega_{6r}(W_A)} p(W|\mathbf{I})} \approx \frac{\frac{p(W_B|\mathbf{I})}{p(W_A|\mathbf{I})}}{\sum_{x_s\in S_{6r}(W_A)} \frac{p(W|\mathbf{I})}{p(W_A|\mathbf{I})}}.$$

Each site $x_s \in S_{6r}(W_A)$ corresponds to a state $W \in \Omega_{6r}(W_A)$. We write the posterior ratio in a factorized form.

$$\frac{p(W|\mathbf{I})}{p(W_A|\mathbf{I})} = \frac{p(T_i)p(T_j)}{p(T_k)} \cdot \frac{p(K^T+1)}{p(K^T)} = \frac{1}{\exp\{-E_T(C_{T_k}(i), C_{T_k}(j))\}} \cdot \frac{p(K^T+1)}{p(K^T)},$$

where $C_{T_k}(i)$ and $C_{T_k}(j)$ are the two curves whose parent-child relation is cut in three T_k leading to two new trees T_i and T_j. Curve $C_{T_k}(i)$ becomes the root curve in tree T_i and curve $C_{T_k}(j))$ becomes a leaf curve in tree T_j. Thus, the proposal probability is computed by

$$\mathbf{Q}_{6r}(W_B|W_A) = \frac{\omega_{6r}(B)}{\sum_{i=1}^{N_{6r}} \omega_{6r}^{(i)}}.$$

$\omega_{6r}^{(B)}$ is the weight for a site $z_{6r}^{(B)} \in \hat{S}_{6r}(W_A)$ that leads to state W_B and it is computed according to the probability by parent-child relation $\exp\{-E_T(C_{parent}(z_B), C_{child}(z_B))\}$ at the site $z_{6r}^{(B)}$. The orders $\{\alpha_1, \alpha 2, ..., \alpha_n\}$ are decided by the tree structure in the parent-child relation directly.

Similarly we update the weights in the candidate set $S_{6l}(W_B)$ and compute the proposal probability $\mathbf{Q}_{6r}(W_A|W_B)$ as the normalized weight.

4.8 Summary on the six simple jumps

The five simple jumps presented so far have a total of 10 sets of "particles" as Fig.12 displays. Each particle is a candidate with its weight approximating the posterior probability ratio in a factorized form. These particles encode the proposal probability in each jump scope and will be re-weighted on-line.

4.9 Jump VII: Grouping/ungrouping of parallel curve groups

The seventh pair of jumps is to group a number of free curves into a parallel curve group pg_i or split a group of free curves from one group pg_i to a group pg_j.

Figure 13 shows a reversible jump between two states

$$W_A = (pg_1^A, pg_2^A, W_-) \rightleftharpoons (pg_1^B, pg_2^B, W_-) = W_B.$$

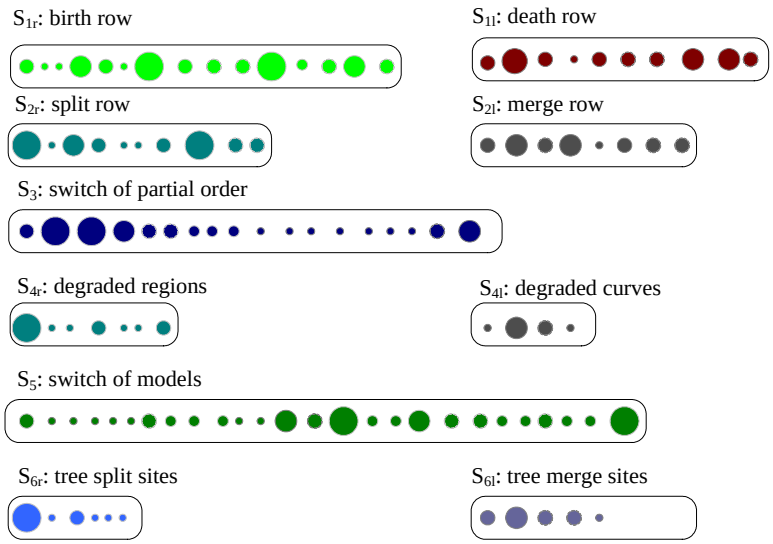


Figure 12: The 6 simple jumps maintain 10 sets of "particles" whose sizes illustrate their weights. The sets are updated and re-weighted in each jump steps, and they encode the proposal probabilities in a non-parametric representation.

W_A has two curve groups – pg_1^A includes 7 curves $(C_1, C_2, C_3, C_4, C_5, C_6, C_8)$ and pg_2^A has three curves (C_7, C_9, C_{10}). Three curves C_1, C_3, C_4 (Fig.13.(a)) are split from pg_1^A and merged with pg_2^A to form two new parallel groups– pg_1^B and pg_2^B (Fig.13.(c)). Each curve group is illustrated by a dotted ellipse.

Suppose we have a set of free curves $\{C_1, C_2, ..., C_N\}$ and we treat each curve as a single vertex and build an adjacency graph G which connects two nearby curves c_s and c_t with a link $e_{st} =< C_s, C_t >$. Fig. 13.(b) shows an example of the adjacency group. Therefore the curve grouping problem becomes a graph coloring or partition problem – all curves (vertices) with the label (or color) belong to a curve group. Thus we adopt a Swendsen-Wang Cut algorithm [2, 3] developed by the authors' group for partition (coloring) this graph G. The SW-cut algorithm is a generalization to the Gibbs sampler. Conventional Gibbs sampler flips the color/label of a single vertex each time and often gets stuck at local minima. In contrast, the SW-cut algorithm can flip a set of vertices that have the the same color at once, and it is shown to converge (mix) much faster than the Gibbs sampler. We call such jumps the "composite jumps".

We briefly introduce the SW-cut idea below and refer to [2] for details. We associate a binary variable b_{st} to each link e_{st} in the adjacency graph G. $b_{st} = 0$ means the link is "off" and thus C_s and C_t are disconnected. $b_{st} = 1$ means the link remains connected. Each link e_{st} is also associated

with a probability q_{st} which measures how likely the two curves C_s and C_t belong to the same curve group.

$$q_{st} \propto \exp\{-E_{pg}(C_s, C_t)\}.$$

The energy E_{pg} measures the distance, parallelism of the two curves and was discussed in eqn. (6).

Each vertex C_s is assigned a label $\ell_s \in \{1, 2, ..., K^{pg}\}$, and curves that have the same label belong to a common parallel group pg whose goodness of fit is measured by a posterior probability $p(pg|\mathbf{I}_{\Lambda_{pg}})$. The number of possible labels K^{pg} is unknown. Suppose we start with every free curve has a different label. A jump in the SW-cut algorithm includes two steps.

Step I: Clustering. For each link e_{st} in G, b_{st} is turned off (set to 0) deterministically if $\ell_s \neq \ell_t$ in the current state W_A. Otherwise b_{st} is turned off with a probability $1 - q_{st}$. That is, b_{st} is an auxiliary variable following the following Bernoulli probability,

$$b_{st} \sim \text{Bernoulli}(q_{st}\mathbf{1}(\ell_s = \ell_t)).$$

This procedure generates a number of connected components (CP) for the adjacency graph G. Each CP has a connected subgraph after turning off a number of links in G. This is called "clustering" of the graph. Each cluster will be a candidate for flipping color. For example, Fig.13.(e) shows 3 CPs (or clusters) which can be generated from both state W_A and state W_B.

Step II: flipping. One connected component is picked at random (e.g. a uniform probability). For example, suppose the current state is W ($W = W_A$ or $W = W_B$) and we pick CP_1 at Fig.13.(e), and suppose the current label of CP_1 is $\ell(CP_1) \in \{1, 2, 3, ..., K^{pg}\}$. We assign a new color $\ell'(CP_1) \in \{1, 2, ..., K^{pg} + 1\}$ to all curves in CP_1 at a proposal probability $q(\ell'(CP_1)|W)$. For example, at state W_A, CP_1 belongs to pg_1^A and $\ell(CP_1) = 1$. At state W_B, CP_1 belongs to pg_2^B and $\ell(CP_1) = 2$. When CP_1 is assigned an existing color $\ell'(CP_1) \in \{1, K^{pg}\}$, it means that CP_1 is split from the current group $pg_{\ell(CP_1)}$ and merged with $pg_{\ell'(CP_1)}$. In a special case when $CP_1 = pg_{\ell(CP_1)}$ is a whole parallel group at the current state, then this merge reduces the number of parallel groups. On the other hand, if CP_1 is assigned a new color $\ell'(CP_1) = K^{pg} + 1\}$, it means that CP_1 becomes a new parallel group.

In Fig. 13, according the the SW-cut algorithm[2, 3], the proposal probability ratio is given by,

$$\frac{\mathbf{Q}(W_B|W_A)}{\mathbf{Q}(W_A|W_B)} = \frac{\prod_{e_{st}\in \text{Cut}_A}(1 - q_{st})}{\prod_{e_{st}\in \text{Cut}_B}(1 - q_{st})} \cdot \frac{q(\ell'(CP_1) = 2|W_A)}{q(\ell'(CP_1) = 1|W_B)}. \tag{36}$$

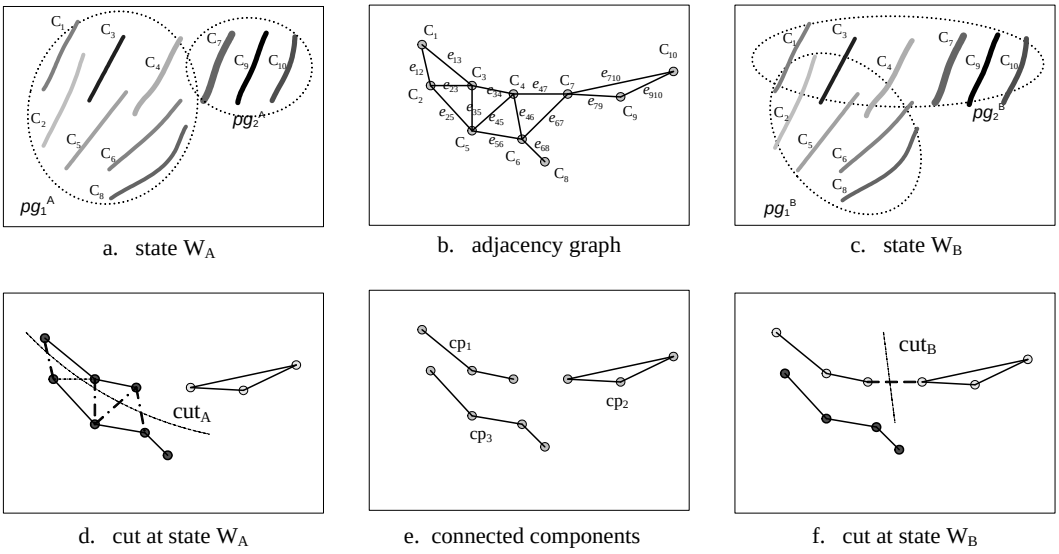


Figure 13: An example of split-merge of parallel curve groups by a composite jump designed with the SW-Ccut method.

In the above equation, Cut_A is the set of links in G at state W_A that connect CP_1 with the rest of pg_1^A They must be cut (turned off) in order for CP_1 being a connected component. So the probability of turning off them in the clustering step is $\prod_{e_{st}\in \text{Cut}_A}((1-q_{st})$. Similarly, Cut_B is the set of links in G at state W_B that connect CP_1 with the rest of pg_2^B. Cut_A and Cut_B are illustrated in Fig.13.(d) and Fig.13.(f) respectively by the dashed lines.

By a Metropolis-Hastings step, the proposed jump is accepted with probability,

$$\alpha(W_A, W_B) = \min(1, \frac{\mathbf{Q}(W_B|W_A)}{\mathbf{Q}(W_A|W_B)} \cdot \frac{p(pg_1^A)p(pg_2^A)}{p(pg_1^B)p(pg_2^B)}). \tag{37}$$

In the above equation, the posterior probability ratio $\frac{p(W_B|\mathbf{I})}{p(W_A|\mathbf{I}}$ is reduced to the ratio $\frac{p(pg_1^A)p(pg_2^A)}{p(pg_1^B)p(pg_2^B)}$ on the prior for the new parallel groups. This is due to two reasons. (1) The individual curves are fixed and thus the likelihood does not change during the jump between W_A and W_B. (2) Once a parallel group is formed, the sequence orders $\alpha_1, \alpha 2, ..., \alpha_n$ of the curves within the group is quite obvious and is thus deterministically decided. In [3], we can choose the proposal probability ratio of the new labels $\frac{q(\ell'(CP_1)=2|W_A)}{q(\ell'(CP_1)=1|W_B)}$ so that the acceptance probability is 1 (i.e. always accepted). But this may not necessarily improve the computational speed.

In summary, the composite jump designed by SW-cut has three properties.

1. Unlike the simple jumps which maintain a list of candidates for its proposals, this composite jump explores a combinatorial number of possible candidates in the clustering step.

2. Like the simple jumps, the proposal probability ratio is also factorized into a number of discriminative probabilities $1 - q_{st}$.

3. It realizes death-birth and split-merge operators of parallel groups depending on the selection of connected component and the assignment of new labels.

5 Experiments

In Fig. 2 we have showed some examples where image segmentation algorithm produces unsatisfactory results because these input images have curve patterns that do not fit the region assumptions. Much improved results are obtained on these images when the curve processes are modeled explicitly, as we shall show in this section.

Our experiments proceed in two stages. We first compute only regions and free curves, and then we run jumps VI and VII to obtain the parallel curve groups and trees.

Experiment A: compute regions and free curves.

Six examples are shown in Fig. 14 and 19. For each example, the first row (from left to right) displays the input image $\mathbf{I}_{\text{obs}}$, the computed free curves W^c, and the region segmentations W^r in the background. The second row shows (from right to left) the synthesized image according to the generative models for the regions $\mathbf{I}^r_{\text{syn}} \sim p(\mathbf{I}|W^r)$, the curves $\mathbf{I}^c_{\text{syn}} \sim p(\mathbf{I}|W^c)$, and the overall synthesis (reconstructed) $\mathbf{I}_{\text{syn}}$ by occluding $\mathbf{I}^c_{\text{syn}}$ on $\mathbf{I}^r_{\text{syn}}$.

We compute the synthesis images to verifies what the computer understands about the input. For example, Fig.14 shows that the computer did not understand the faces of the three Japanese ladies and treated them as regions.

In these experiments, two parameters in the prior models are adjustable: (1) γ_r in equation (11), and (2) γ_c in equation (4). The two parameters control the extent of the segmentation, i.e. the number of regions and curves. Therefore they decide how detailed we like the parsing to be. Usually we set $\gamma_r = 5.0$ and $\gamma_c = 3.5$ and other parameters are fixed.

Experiment B: assuming regions, curves and parallel groups, and trees.

In the second experiment, we further compute the parallel groups and trees by turning on the two composite jumps $\mathbf{J}_7, \mathbf{J}_8$. Figures 20 to 23 show four examples. In each example, the top row shows the parallel groups or trees grouped from the simple curves. The second and third rows are displayed as before. ¿From the results, we can see that the algorithm successfully segments,

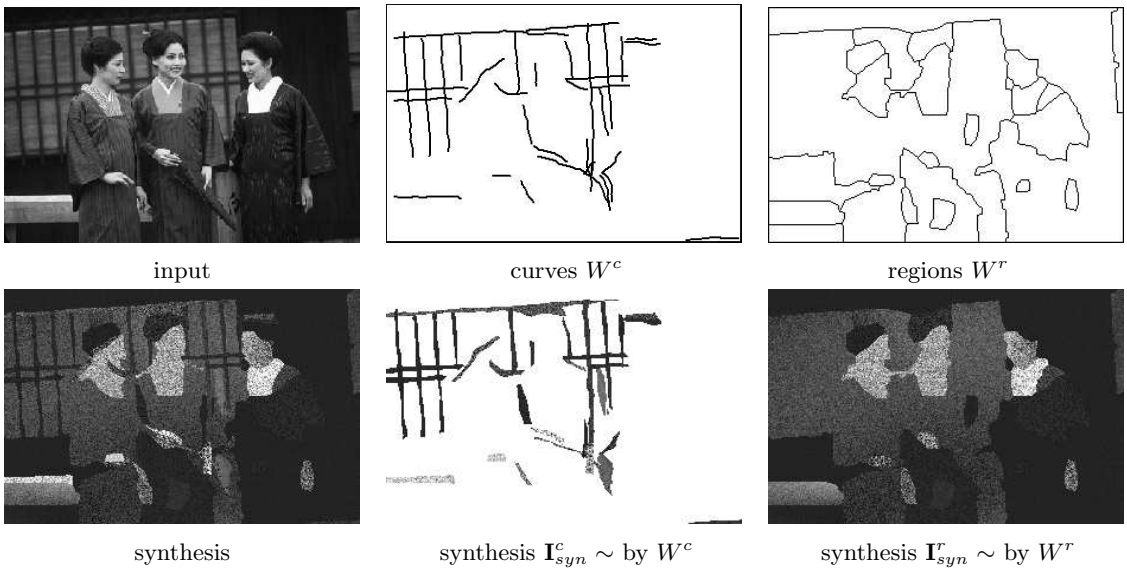


Figure 14: Experiment A1: parsing images into regions and free curves.

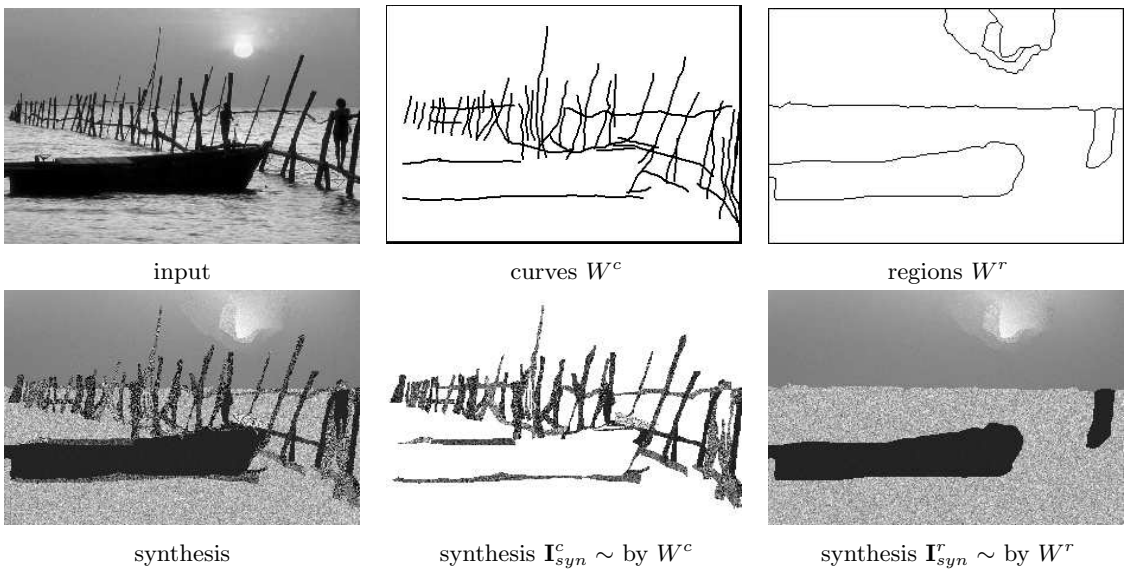


Figure 15: Experiment A2: parsing images into regions and free curves.

detects, and groups regions, curves, and curve groups respectively.

We observed some problems with the zebra image in Fig.21. There are simple curves computed for both black and white stripes. The prior model for parallel groups emphasizes parallelism not intensity similarity, thus the stripes are divided into three parallel groups.

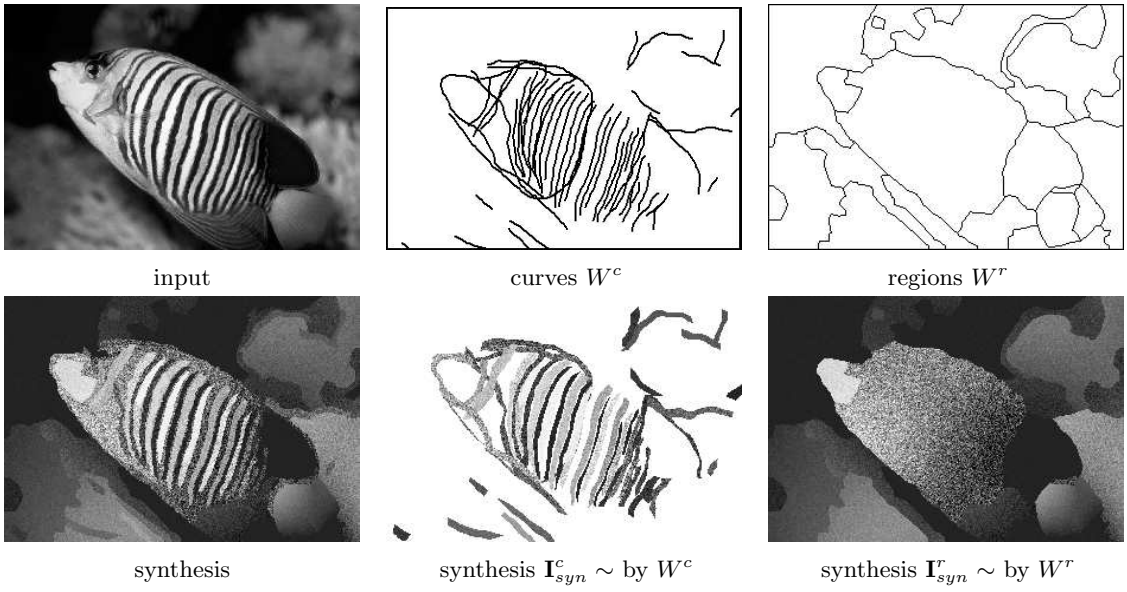


Figure 16: Experiment A3: parsing images into regions and free curves.

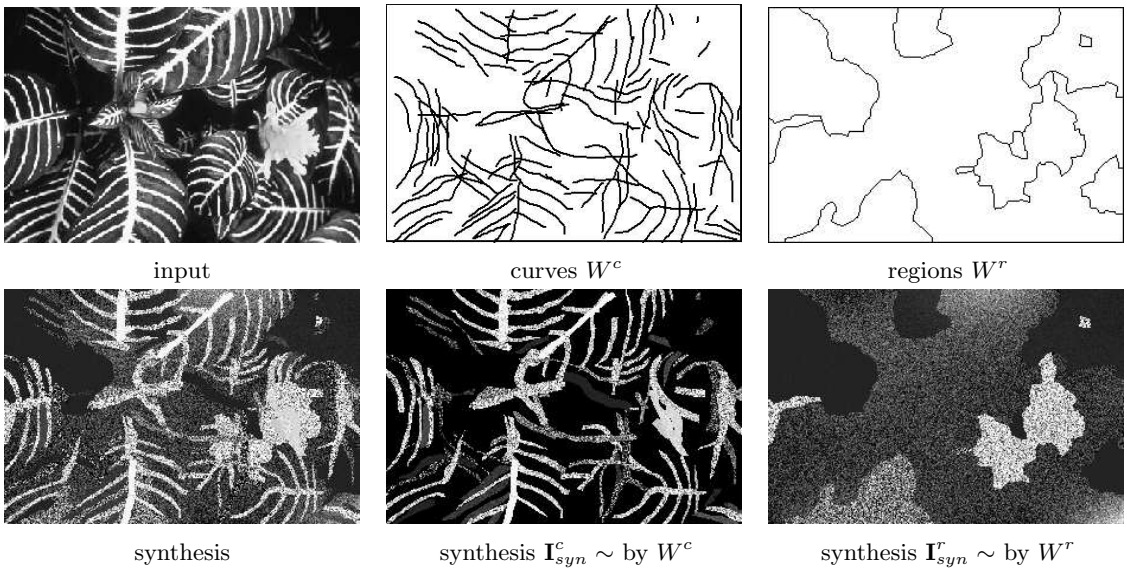


Figure 17: Experiment A4: parsing images into regions and free curves.

Computational time. It usually takes 20 minutes for the algorithm to produce a result on an image of size 300×200 pixels, because of the integration of region and curve models. The code is not well structured at the moment as it is incrementally added over time. We expect to bring the computing time to within 1 minute per image on regular PCs.

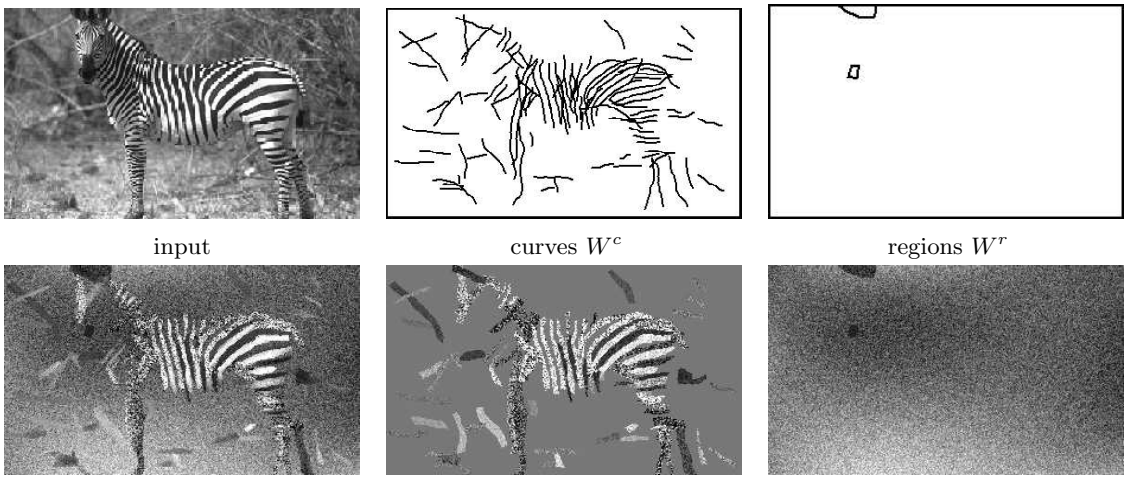


Figure 18: Experiment A5: parsing images into regions and free curves.

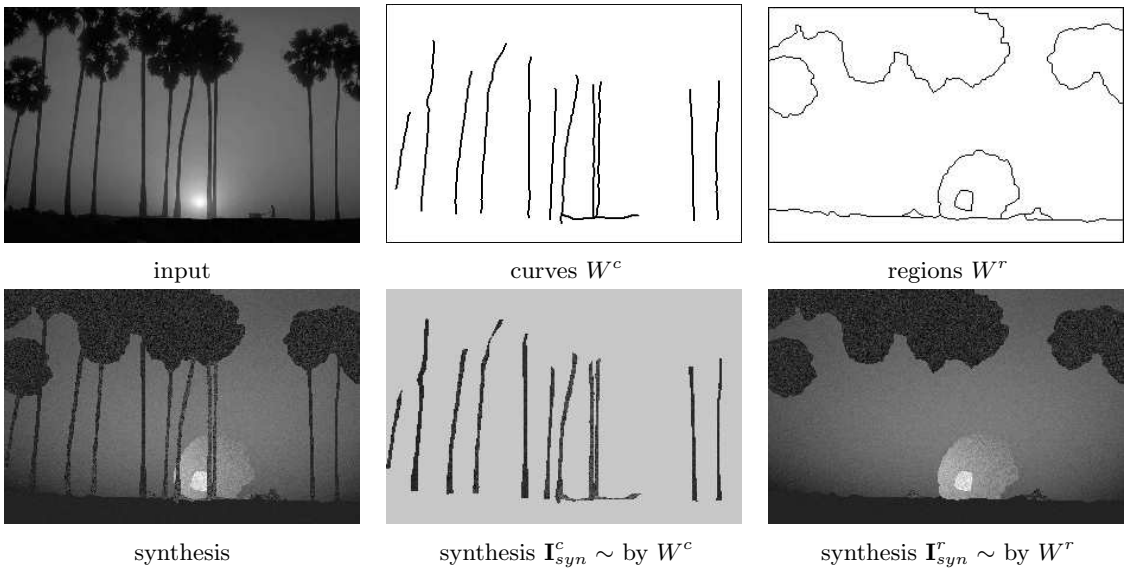


Figure 19: Experiment A6: parsing images into regions and free curves.

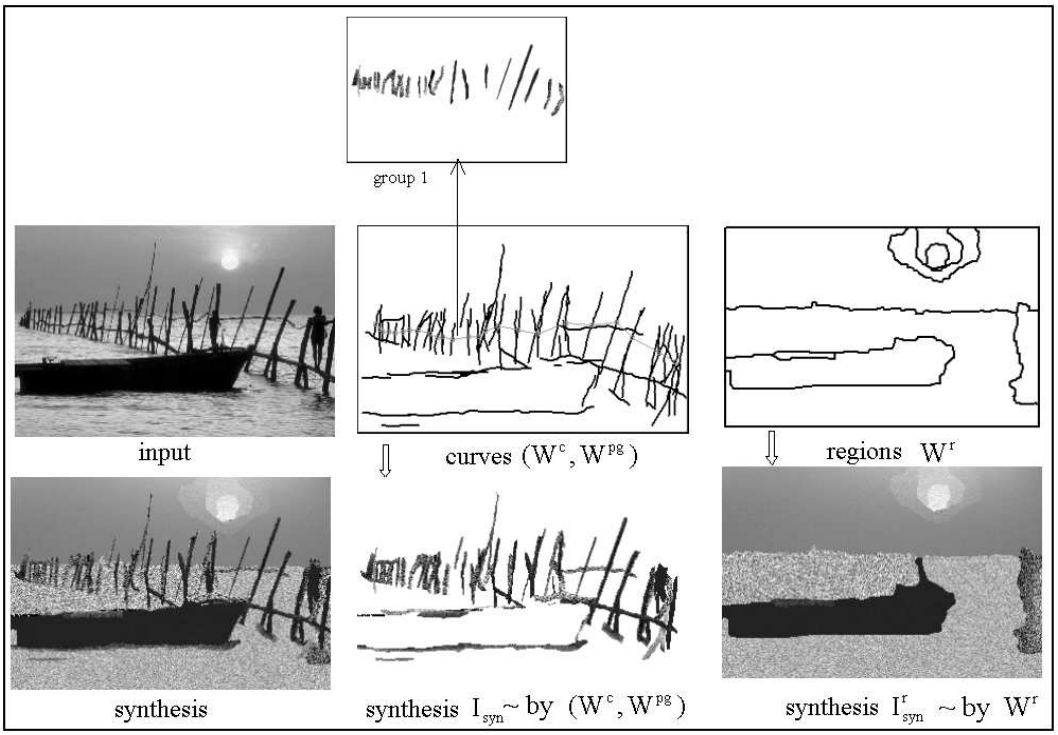


Figure 20: Experiment B1: parsing an image into regions, curves, and parallel curve groups.

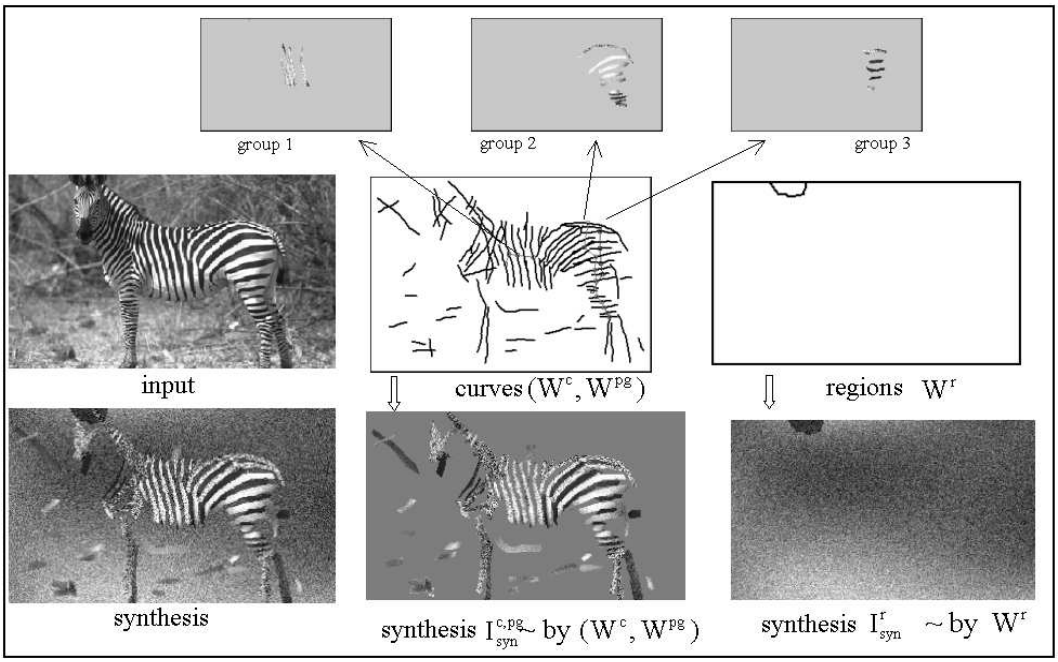


Figure 21: Experiment B2: parsing an image into regions, curves, and parallel curve groups.

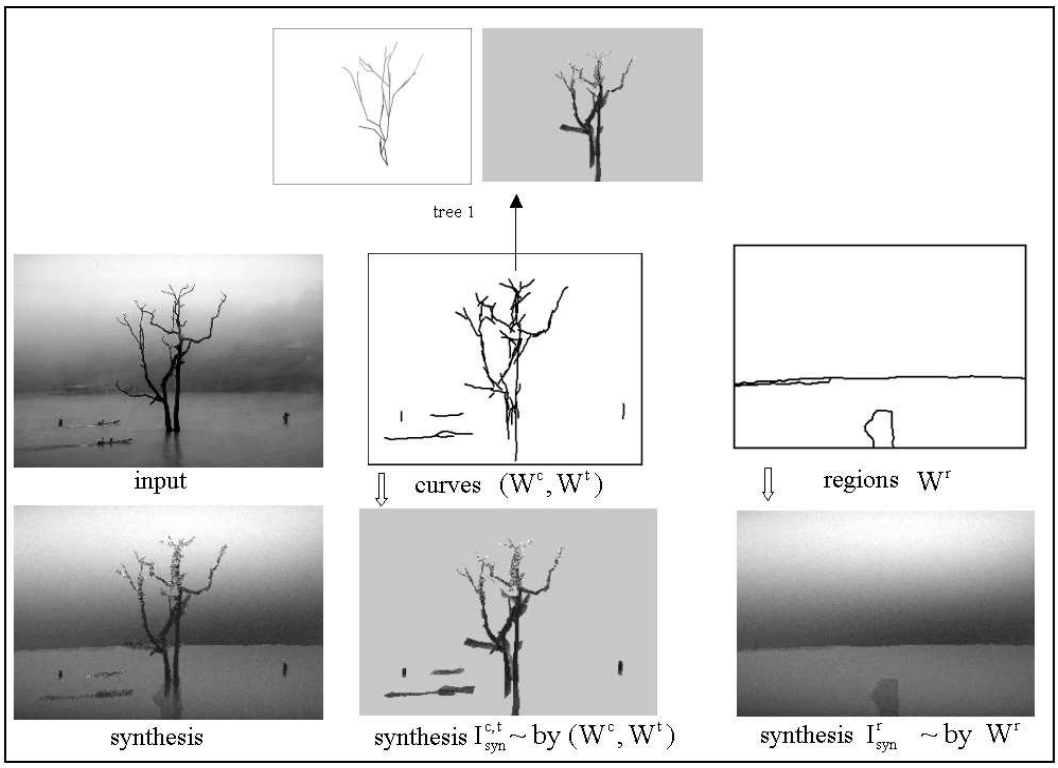


Figure 22: Experiment B3: parsing a tree image into regions, curves, and trees

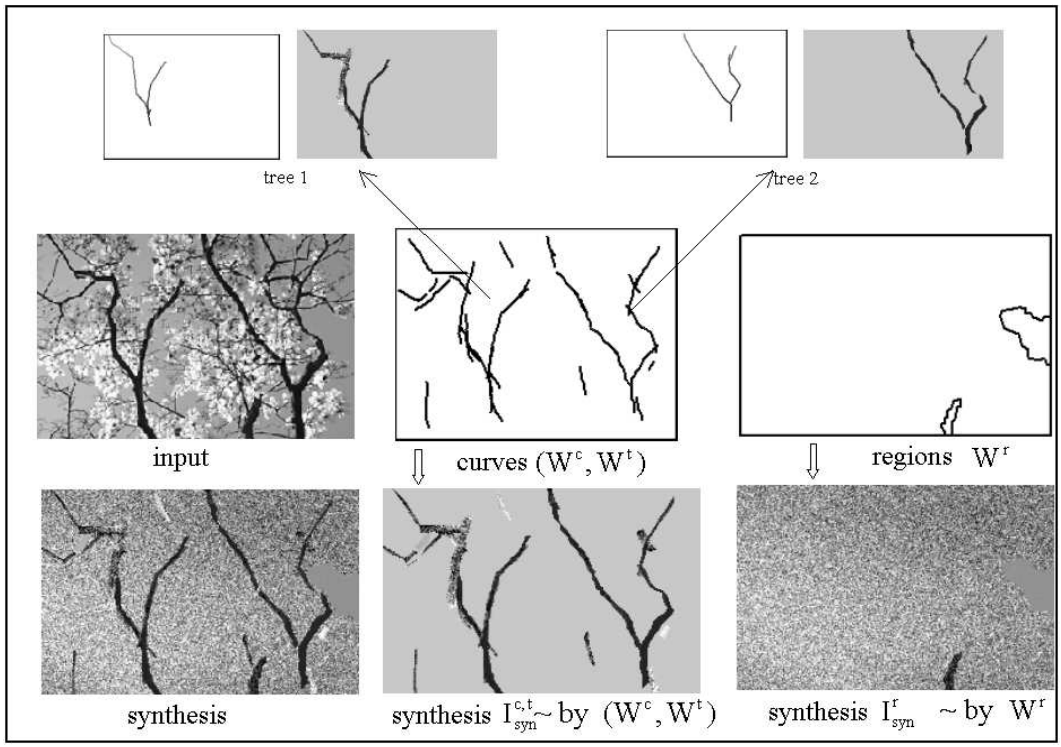


Figure 23: Experiment B4: parsing an image into regions, curves, and trees.

6 Discussion and future work

In this paper, we introduce a MCMC method for solving two middle level vision tasks together: image segmentation and curve detection. Three aspects are crucial to the integrated solution. The first aspect is the use of generative models and priors to encode a variety of regularities in a Bayesian formulation. The generative representation enables the curve structures and regions compete to explain the image. The second aspect is the design of Markov chain transition kernel. It is composed of seven pairs of reversible jumps plus other jumps for the region segmentation. These jumps can traverse the state space. The third aspect is to use discriminative models for composing the proposal probabilities which approximate the posterior probabilities ratio in factorized forms.

This paper is part of a series of work integrating discriminative and generative models from low-level image segmentation [37], middle-level curve structures detection [38], to high-level object recognition [39]. The representation computed in this work has been used in reconstructing 3D scene from a single image [20].

Acknowledgment

This work is supported by an NSF grant IIS-02-44763 and an ONR grant N000140-02-1-0952. We thank Adrian Barbu and Romeo Maciuca for discussions and assistance. We thank Dr. Alan Yuille for general support and many insightful discussions.

References

[1] J. August and S. W. Zucker, "Sketches with Curvature: The Curve Indicator Random Field and Markov Processes", *IEEE Trans. PAMI*, vol. 25, no. 4, April, 2003.

[2] A. Barbu and S.C. Zhu, "Graph Partition by Swendsen-Wang Cuts", *Proc. of ICCV*, Nice, France, 2003.

[3] A. Barbu and S.C. Zhu, "Cluster Sampling and Its Applications in Image Analysis", *Preprint 409* Department of Statistics, UCLA, 2005.

[4] P. Bremaud, "Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues," Springer, 1999. (chapter 6).

[5] R. Bubley and M. Dyer, "Path Coupling: a Technique for Proving Rapid Mixing in Markov Chains", *Proc. of 38th Annual IEEE Symposium on Foundations of Computer Science*, 1997.

[6] E. J. Candes, "Ridgelets: Theory and Applications," *Ph.D. Dissertation*, Yale University, 1998.

[7] S. Carlsson, "Sketch Based Coding of Grey Level Images," *Signal Processing*, vol. 15, 1998.

[8] D. Comaniciu and P. Meer, "Mean Shift Analysis and Applications," *Proc. of ICCV*, 1999.

[9] C. Cooper and A.M. Frieze, "Mixing Properties of the Swendsen-Wang Process on Classes of Graphs", *Random Structures and Algorithms* 15, 242-261, 1999.

[10] T. F. Cootes, D. Cooper, C.J. Taylor and J. Graham, "Active Shape Models - Their Training and Application," *Computer Vision and Image Understanding*, vol.61, no. 1, Jan. pages 38-59,1995.

[11] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun, "EM, MCMC, and Chain Flipping for Structure from Motion with Unknown Correspondence", *Machine Learning*, no. 50, pp. 45-71, 2003.

[12] F. Han, Z. W. Tu, and S.C. Zhu, "Range Image Segmentation by an Effective Jump-Diffusion Method", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp 1138-1153, 2004.

[13] A. R. Dick, P.H.S. Torr, and R. Cipolla, "A Bayesian estimation of building shaping using MCMC", *Proc. of 6th European Conference on Computer Vision*, 2002.

[14] D.A. Forsyth, "The Joy of Sampling", *Int'l J. of Computer Vision*, 41(1/2), 109-134, January 2001.

[15] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. PAMI*, vol. 6, pp. 721-741, Nov. 1984.

[16] D. Geman and B. Jedynak, "An Active Testing Model for Tracking Roads from Satellite Images," *IEEE Trans. PAMI*, vol. 18, no. 1, 1996.

[17] L. Parida, D. Geiger, and R. Hummel, "Junctions: Detection, Classification, and Reconstruction," *IEEE Trans. PAMI*, vol. 20, no. 7, 1998.

[18] P. J. Green, "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination," *Biometrika*, vol. 82, no. 4, pp. 711-732, 1995.

[19] U. Grenander and M. I. Miller, "Representations of Knowledge in Complex Systems," *J. of the Royal Stat. Soc. Series B*, vol. 56, no. 4, pp.549-603, 1994.

[20] F. Han and S.C. Zhu, "Bayesian Reconstruction of 3D Shapes and Scenes from a Single Image", *Proc. of Int'l workshop on High Level Knowledge in 3D Modeling and Motion*, Nice France, 2003.

[21] W.K. Hastings, "Monte Carlo Sampling Methods Using Markov Chains and Their Applications", *Biometrika*, 57, 97-109, 1970.

[22] M. Isard and A. Blake, "Contour Tracking by Stochastic Propagation of Conditional Density," *Proc. of ECCV*, 1996.

[23] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J. Computer Vision*, vol. 1, no. 4, pp. 321-332, Jan. 1988.

[24] A.D. Lanterman, "Jump-Diffusion Algorithm for Multiple Target Recognition using Laser Radar Range Data", *Optical Engineering*, Vol. 40, No. 8, pp. 1724-1728, 2001.

[25] J.S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer-Verlag NY INC. 2001.

[26] R. Maciuca and S.C. Zhu, "First-hitting-time Analysis of Independence Metropolis Sampler", *Journal of Theoretical Probability* (Accepted), 2005.

[27] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and Texture Analysis for Image Segmentation," *Int'l J. Comp. Vis.*, June 2001.

[28] S.G. Mallat and Z. Zhang, "Matching Pursuits With Time-frequency Dictionaries," *IEEE Tranc. on Signal Processing*, vol. 41, no. 12, Dec. 1993.

[29] N. Metropolis, M.N. Rosenbluth, A.W. Rosenbluth, A.H. Teller, and E. Teller, "Equations of State Calculations by Fast Computing Machines", *J. Chem. Phys.* 21, 1087-92, 1953.

[30] M. Nitzberg and D.B. Mumford, "The 2.1D Sketch", *Proc. of ICCV*, pp.138-144, 1990.

[31] P.H. Puskun, "Optimum Monte Carlo sampling using Markov chains", *Biometrika*, 60(3):607-612, 1973.

[32] R. E. Schapire, "The boosting approach to machine learning: an overview", *MSRI Workshop on Nonlinear Estimation and Classification*, 2000.

[33] S. Skiena, "Partial Orders," *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, MA: Addison-Wesley, pp. 203-209, 1990.

[34] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. PAMI*, vol. 22, no. 8, Aug. 2000.

[35] A Srivastava, U. Grenander, G.R. Jensen, and M. I Miller, "Jump-diffusion Markov Processes on Orthogonal Groups for Object Post Estimation", *J. of Statistical Planning and Inference*, 103, 15-37, 2002.

[36] R. H. Swendsen and J. S. Wang, "Nonuniversal Critical Dynamics in Monte Carlo Simulation," *Physical Review Letters*, vol. 58, no. 2, pp. 86-88, 1987.

[37] Z. Tu and S. C. Zhu, "Image Segmentation by Data-Driven Markov Chain Monte Carlo," *IEEE Trans. PAMI,* Vol. 24, No.5, May, 2002.

[38] Z. Tu and S.C. Zhu, "Parsing Images into Region and Curve Processes," Proc. of *7th European Conference on Computer Vision*, Copenhagen, May 2002.

[39] Z. Tu, X. Chen, A.L. Yuille, and S.C. Zhu, "Image Parsing: Segmentation, Detection and Recognition", *Proc. of ICCV*, Nice, France, 2003.

[40] J. Wang and E. Adelson, "Representing Moving Images with Layers", *IEEE Trans. on Image Processing*, vol.6, no.3, 1994.

[41] S. C. Zhu and A. L. Yuille, "Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multi-band Image Segmentation," *IEEE Trans. PAMI*, vol. 18, no. 9, pp. 884-900, Sept. 1996.

[42] 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 Int'l Conf. on Computer Vision and Pattern Recognition*, Hilton Head, SC. 2000

[43] C. Zimmer, E. Labruyere, V. Meas-Yedid, N. Guillen, J.C. Olivo-Marin, "Segmentation and Tracking of Migrating Cells in Videomicroscopy With Parametric Active Contours" *IEEE Trans. Medical Imaging*, vol. 21, no. 10, 2002.