International Journal of Computer Vision 69(2), 223–249, 2006

DOI: 10.1007/s11263-006-6995-9

Parsing Images into Regions, Curves, and Curve Groups

ZHUOWEN TU
Lab of Neuro Imaging, Department of Neurology, University of California, Los Angeles, CA 90095
ztu@stat.ucla.edu

SONG-CHUN ZHU
Departments of Statistics and Computer Science, University of California, Los Angeles, CA 90095
sczhu@stat.ucla.edu

Received June 4, 2003; Revised December 26, 2005; Accepted December 29, 2005

First online version published in May, 2006

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). This algorithm is targeted for an integrated solution to image segmentation and curve grouping through Bayesian inference. The paper makes the following contributions. (1) It adopts a layered (or 2.1D-sketch) representation integrating both region and curve models which compete to explain an input image. The curve layer occludes the region layer and curves observe a partial order occlusion relation. (2) A Markov chain search scheme *Metropolized Gibbs Samplers* (MGS) is studied. It consists of several pairs of reversible jumps to traverse the complex solution space. An 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-Hastings step. This paper discusses systematic design strategies of devising reversible jumps for a 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 they 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

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). This 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) (Nitzberg and Mumford, 1990; Wang and Adelson, 1994) representation illustrated in Fig. (1). A region is a two-dimensional compact area with coherent intensity patterns. 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 solely refer curves as the

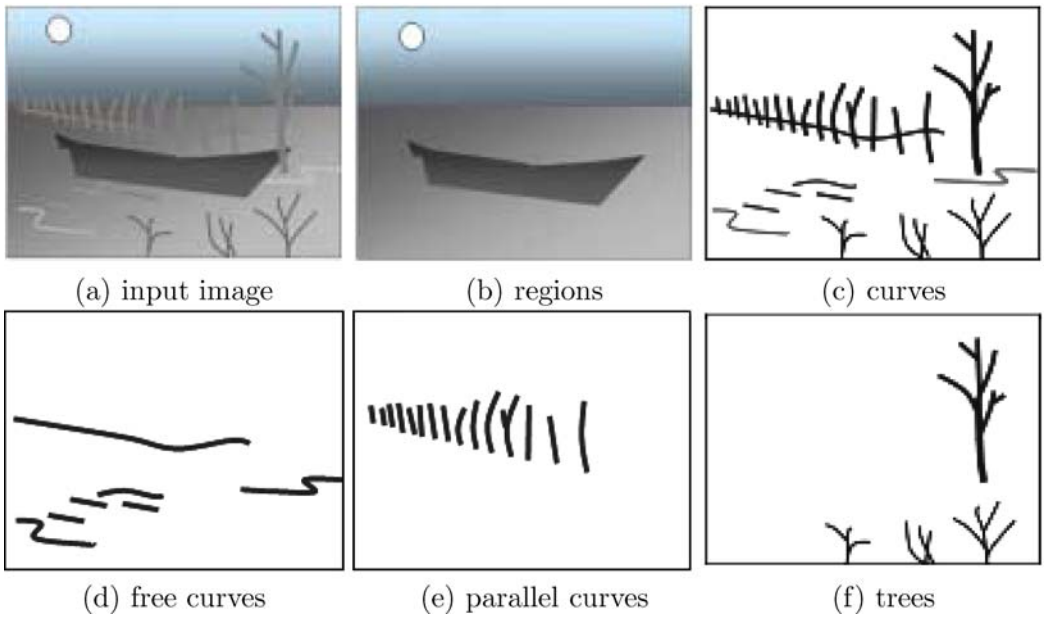


Figure 1. An illustration of parsing an image into regions, curves, and curve groups. (a) is an input image which is decomposed into two layers, (b) a layer of regions and (c) a layer of curves. These curves are further divided into (d) free curves, (e) a parallel curve group for the fence, and (f) trees.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

boundaries of 2D regions (Kass et al., 1988; Isard and Blake, 1996; Malik et al., 2001). 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 as Markov tree structures. Curve structures are assumed to observe a partial-order occlusion relation and they all occlude the region layer.

While there is a wealthy body of work on *image segmentation* and *curve detection/grouping* respectively, these 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 (Tu and Zhu, 2002) and the curves make the segmentation rather cluttered. For comparison Fig. (14)–(23) demonstrate significantly improved results when the curve structures are represented and computed separately. On the other hand, curve detection and grouping algorithms often lack models for background regions, and thus assume uniform background, or as an alternative, they have to adopt discriminative curve models that work on the differences between curves and background.

Given an input image, our objective is to infer an unknown number of regions, free curves, parallel groups, and trees, with recovered occlusion relation and their probability models selected and fitted—all in the process of maximizing (or simulating) a Bayesian posterior probability. This 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.

This paper is mainly focused on studying a systematic search strategy in such a complex state space 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. This is done by designing a number of pairs of jumps to form

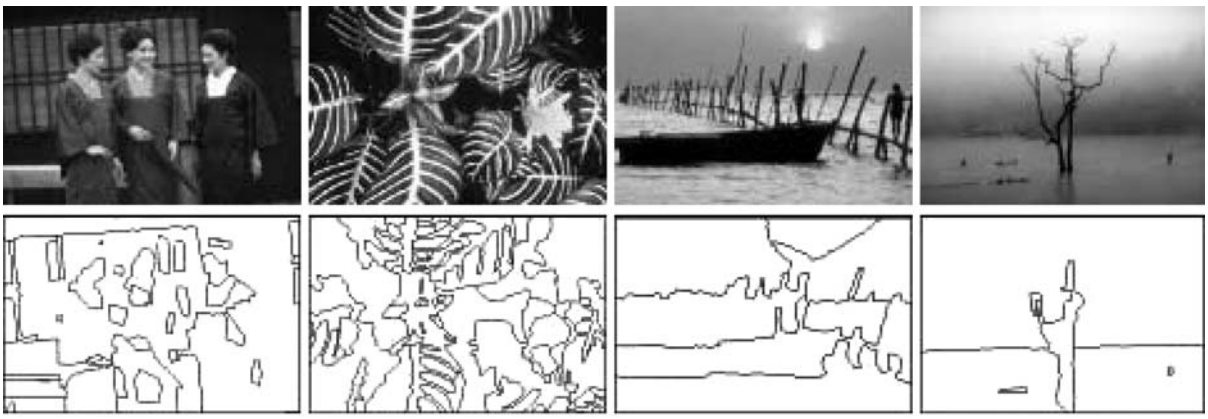


Figure 2. Degraded results in image segmentation in the presence of curves. In comparison, Figures (14–23) show much improved results when the curve structures are represented and computed.

an ergodic Markov chain. The resulting Markov chain can reach any states from an arbitrary initialization.

Secondly, each 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. This is like a Gibbs sampler. The proposal is then accepted in a Metropolis-Hastings step, hence its name, the *Metropolized Gibbs Sampler* (MGS Liu (2001)).

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 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.

1.2. Relation to Previous Work and Alternative Methods

1.2.1. Relation to Previous Work on Markov Chain Monte Carlo. Stochastic computing with reversible jumps was pioneered in Grenander and Miller (1994) and Green (1995). A data-driven Markov chain Monte Carlo (DDMCMC) framework (Tu and Zhu, 2002; Zhu et al., 2000) was proposed to improve the speed of reversible jumps by computing the proposal probabilities with factorized discriminative models. The DDMCMC framework was originally illustrated in image segmentation. Lately, this framework has been applied to integrating high level vision tasks such as face and text detection with the segmentation process (Tu et al., 2003). The method presented in this paper has been extended in Han and Zhu (2003) to 3D scene reconstruction from a single image based on the region and curve representation. In this paper we focus on the curve detection and grouping task, its interactions with image segmentation, and a general design principle of MGS. 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 (Dellaert et al., 2003; Forsyth, 2001), 3D modeling (Dick et al., 2002), object recognition (Lee and Cohen, 2004) and tracking (Zhao and Nevita, 2004; Khan et al., 22004). 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 to facilitate the design of MCMC algorithms.

1.2.2. 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) (Kass et al., 1988), road tracing in satellite images (Geman and Jedynak, 1996),

medical image analysis (Zimmer et al., 2002), object tracking (Isard and Blake, 1996), curve reconstruction from multiview images (Kaess et al., 2004), and image coding using ridgelets and curvelets (Candes, 1998; Carlsson, 1998). Existing methods have various restrictive assumptions. (1) Many methods require manual initialization of the curve near the right position (Kass et al., 1988; Isard and Blake, 1996; Cootes et al., 1995) or manually initializing the starting point for tracing (Geman and Jedynak, 1996). (2) Most algorithms assume uniform background (August and Zucker, 2003; Zimmer et al., 2002) since they lack generative models for them. 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 (Candes, 1998; Carlsson, 1998) assume a generative model that images are linear addition of curve elements/bases. Our early attempt (Tu and Zhu, 2002) adopted this additive model with the image bases organized in a Markov chain. The additive model results in artifacts, such as blurry boundaries while improved results in this paper using occlusion. (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.

1.2.3. Comparison with Alternative Methods. The Markov chain jumps can be considered as generalizations to conventional gradient descent moves in three aspects. (1) 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. (2) A jump can move in a rather large scope at a single step, while gradient methods move within a small local neighborhood. (3) 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 in a Bayesian framework. 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

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

2.1.1. 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 $\mathbf{c}_r = (x_r(s), y_r(s))$. Figure 3(a) shows the boundaries in dashed lines.

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

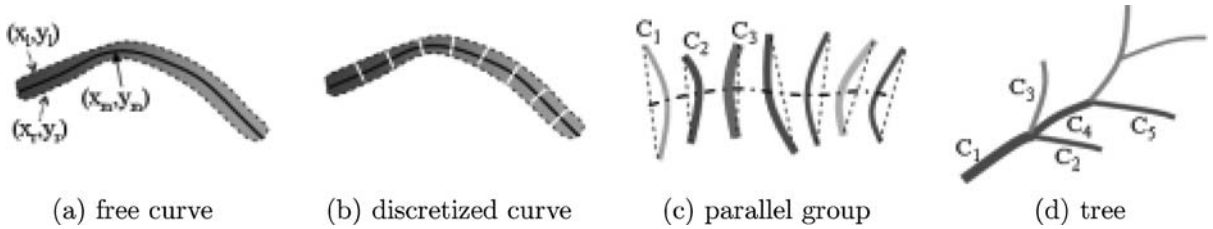


Figure 3. Representations of curves and curve groups. (a) A free curve in continuous representation. (b) A free curve is discretized into a chain of "bars". (c) Curves for a parallel group (d) Curves for a Markov tree.

where $\mathbf{n}(s)$ is the unit normal of $\mathbf{c}_m(s)$. Intuitively, a curve is a degenerated region parameterized by its 1D medial axis. Usually $w(s)$ is only 1–3 pixels wide and $w \ll L$. This causes major topology problems in image segmentation where the two boundaries $\mathbf{c}_l(s)$ and $\mathbf{c}_r(s)$ could often intersect generating numerous trivial regions. This problem will be resolved with the explicit 1D representation. The intensities of a curve often exhibit globally smooth shading patterns, for example the curves in Figs. (14)–(23). Thus we adopt a quadratic function for curve intensities,

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

with parameters $\theta_0 = (a, b, c, d, e, f)$. The validation of choosing an inhomogeneous model to capture the smoothly changing intensity patterns can be found in Tu and Zhu (2002). Therefore, a free curve is described by the following variables in continuous representation

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

where σ is the variance of the intensity noise. While this continuous representation is a convenient model, we should also work on a discrete representation. Then the domain $\mathcal{D}(C)$ is a set of pixels in a lattice and C is a chain of elongated bars as Fig. 3(b) illustrates.

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

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

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

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

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 $\mathcal{N}(0; \sigma^2)$. The image likelihood therefore is

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

2.1.2. Parallel Curve Groups. A parallel curve group consists of a number of nearly parallel curves as Fig. 3(c) shows. Each curve $C_i, i = 1, 2, \ldots, n$ is summarized by a short line segment connecting its end points. They represent curve structures, such as zebra stripes, grids, and railings shown in the experiments. Grouping curves into a parallel group is encouraged in the model as it reduces coding length and it is useful for perceiving an object, for example, a zebra. We denote a parallel curve group by

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

$\alpha_i \in \{1, \ldots, n\}$ 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 as

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

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.

2.1.3. Markov trees. Figure 3(d) shows a number of curves in a Markov tree structure. We denote it by

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

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

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

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

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

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

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

2.2. Generative Models of Regions

Once the curves explain away the elongated patterns, what is left within each image are the regions in the background. In this paper, we adopt two simple region models in comparison to the four models in Tu and Zhu (2002). We denote a 2D region by $R \subset \Lambda$ and $\mathbf{I}_R$ the intensities inside R.

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

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

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.

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

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. This is to impose an inhomogeneous model for capturing the gradually changing intensity patterns, e.g. the sky. This model is important since regions with shading effects will be segmented into separate pieces with homogeneous models. The parameters are $\theta = (M, \mathcal{H})$ and more details with a validation can be found in Tu and Zhu (2002) where we compare different models for different types of images.

The likelihood probability is

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

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

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

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 $A = (C_1, C_2, \ldots, C_N)$. We then define a partially ordered set, *poset* (Skiena, 1990), $\mathcal{PR} = \langle A, \preceq \rangle$. $b \preceq c$ means that curve b occludes curve c or b is on top of c. $\mathcal{PR}$ is represented by a directed acyclic graph called a 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 $A = \{a, b, c, d, e, f\}$.

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 (Tu and Zhu, 2002) 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.

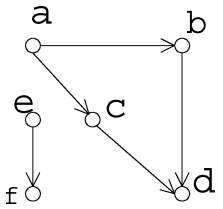


Figure 4. A Hasse diagram for a partial order relation.

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 $\Lambda_R = \mathcal{D}(R) - \cup_{C \in A} \Lambda_C$.

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

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

Similarly, we have $W^c = (K^c, C_1, \ldots, C_{K^c})$, $W^{pg} = (K^{pg}, pg_1, pg_2, \ldots, pg_{K^{pg}})$, and $W^t = (K^t, T_1, T_2, \ldots, T_{K^t})$. In this model, there is no need to define the background since each pixel either belongs to a region or is explained by a curve/curve group.

The problem is posed as Bayesian inference in a solution space Ω.

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

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

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

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

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

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

Since all objects use generative models 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 $p(W \mid \mathbf{I})$ by sampling W in the solution space with a fast simulated annealing procedure. Since W^* is usually highly peaked, we hope that it will most likely be sampled if the algorithm converges to the target distribution. 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, and curve groups in an image. Especially our objective is to achieve automatic 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 Tu and Zhu (2002) for details of image segmentation by DDMCMC. We focus on the analysis and design of Metropolized Gibbs Sampler (MGS) and its approximation. We give a 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. 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 $p(W \mid \mathbf{I})$ in the solution space Ω by a sampling strategy. Note that W has both discrete and continuous random variables and both can be sampled with the reversible jumps(Green, 1995). In practice, diffusion processes are added to the reversible jumps (Grenander and Miller, 1994; Lanterman, 2001; Srivastava et al., 2002) 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 (Zhu and Yuille, 1996).

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 $p(W \mid \mathbf{I})$, i.e.

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

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

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

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

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

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, and grouping-ungrouping.

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.

$$\mathbf{K}_m(W_A, W_B) = \omega_{mr}\mathbf{K}_{mr}(W_A, W_B) + \omega_{ml}\mathbf{K}_{ml}(W_A, W_B). \tag{18}$$

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

$$\mathbf{K}(W_A, W_B) = \sum_{m=1}^{\mu} \omega_m \mathbf{K}_m(W_A, W_B), \quad \omega_1 + \cdots + \omega_\mu = 1, \tag{19}$$

where $\omega_m, m = 1, 2, \ldots, \mu$ 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.

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 *scope* of a jump as the following.

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

$$\Omega_{mr}(W) = \{W' : \mathbf{K}_{mr}(W, W') > 0, W' \in \Omega,\ W' \neq W\},$$

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

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

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

Figure 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.

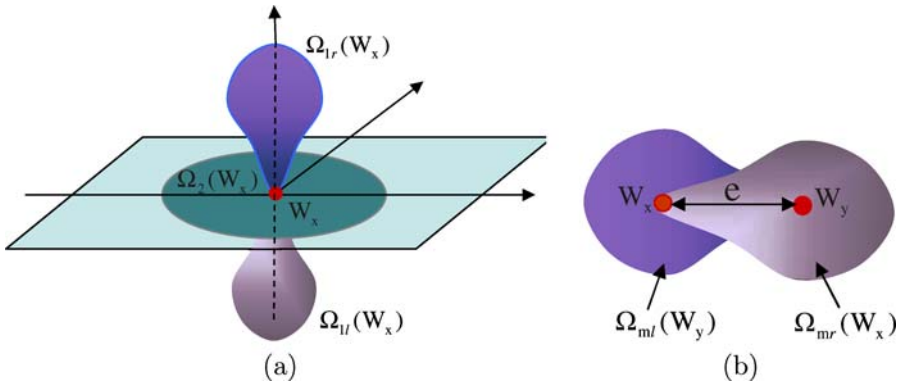


Figure 5. (a) Illustrations for the scope $\Omega(W)$ at a state W. The round domain on the horizontal plane represents the scope of a pair of *symmetric jumps* and the dumb-bells represent the left and right scopes of a pair of *asymmetric jumps*. (b). A reversible jump $\mathbf{J}_m$ between two states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

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

$$W_B \in \Omega_{mr}(W_A) \text{ if and only if } W_A \in \Omega_{ml}(W_B), \quad \forall m. \tag{20}$$

Figure 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 are the jumps for switching image models of a region in the next section where the scope is the image model space 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

$$W_B \in \Omega_{mr}(W_A) \cup \{W_A\} = \Omega_{ml}(W_B) \cup \{W_B\} \ni W_A. \tag{21}$$

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

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

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 (Geman and Geman, 1984) to design $\mathbf{K}_m$ which will observe the invariance Eq. (16). Because $\mathbf{J}_m$ is symmetric (see Eq.(21)), the normalization 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 Eq. (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 (Metropolis et al., 1953; Hastings, 1970).

$$\mathbf{K}_m(W_A, W_B) = \mathbf{Q}_{mr}(W_A, W_B)\alpha_{mr}(W_A, W_B), \quad \text{for } W_A \neq W_B,\ m = 1, 2, \ldots, \mu. \tag{23}$$

$\mathbf{Q}_{mr}(W_A, W_B) = \mathbf{Q}_{mr}(W_B \mid W_A)$ 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,

$$\alpha_{mr}(W_A, W_B) = \min\left(1, \frac{\mathbf{Q}_{ml}(W_A \mid W_B)}{\mathbf{Q}_{mr}(W_B \mid W_A)} \cdot \frac{p(W_B \mid \mathbf{I})}{p(W_A \mid \mathbf{I})}\right). \tag{24}$$

It uses the target probability ratio $\frac{p(W_B \mid \mathbf{I})}{p(W_A \mid \mathbf{I})}$ to rectify the proposal probability ratio $\frac{\mathbf{Q}_{ml}(W_A \mid W_B)}{\mathbf{Q}_{mr}(W_B \mid W_A)}$. Thus,

$$\mathbf{K}_{mr}(W_A, W_A) = 1 - \sum_{W_B \neq W_A} \mathbf{K}_{mr}(W_A, W_B), \ \forall W_A \in \Omega. \tag{25}$$

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.

Let us consider 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.

$$\mathbf{Q}^*_{mr}(W_B \mid W_A) = \frac{p(W_B \mid \mathbf{I})}{\sum_{W_C \in \Omega_{mr}(W_A)} p(W_C \mid \mathbf{I})}, \quad \text{for } W_B \in \Omega_{mr}(W_A) \tag{26}$$

$$\mathbf{Q}^*_{ml}(W_A \mid W_B) = \frac{p(W_A \mid \mathbf{I})}{\sum_{W_C \in \Omega_{ml}(W_B)} p(W_C \mid \mathbf{I})}, \quad \text{for } W_A \in \Omega_{ml}(W_B). \tag{27}$$

We set $\mathbf{Q}^*_{mr}(W_B \mid W_A) = 0$ and $\mathbf{Q}^*_{ml}(W_A \mid W_B) = 0$ outside the two scopes respectively.

The proposal probability is the same as the Gibbs sampler except that we set $\mathbf{Q}^*_m(W_A, W_A) = 0, \ \forall W_A$ (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 $p(W_B \mid \mathbf{I})$. Thus it generates fair samples rapidly.

This design is called the *Metropolized Gibbs sampler* (MGS) following a simple example in Liu (2001), 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 jump scopes and good proposal probabilities, and the latter requires fast computation of the proposal probabilities.

In Eqs. (26) and (27), the MGS proposals $\mathbf{Q}^*_{mr}(W_x \mid W_y)$ and $\mathbf{Q}^*_{ml}(W_y \mid W_x)$ compute the target probability $p(W_C \mid \mathbf{I})$ over two scopes $W_C \in \Omega_{mr}(W_A) \cup \Omega_{ml}(W_B)$. We observe that

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

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

While it is hard to compute $p(W_C \mid \mathbf{I})$) for every state $W_C \in \Omega_{mr}(W_A) \cup \Omega_{ml}(W_B)$ at each step, it is much easier to compute the ratio $\frac{p(W_C \mid \mathbf{I})}{p(W_A \mid \mathbf{I})}$ or $\frac{p(W_B \mid \mathbf{I})}{p(W_A \mid \mathbf{I})}$ since W_A and W_C differ in just 1-2 items. Most of the terms are thus canceled when we compute the ratios.

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. Thus, the detailed balance equations are still observed.

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

each candidate.

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

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,

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

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

$$\mathbf{K}_{mr}(W_A, W_B) = \mathbf{Q}_{mr}(W_B \mid W_A) \min\left(1, \frac{\mathbf{Q}_{ml}(W_A \mid W_B)}{\mathbf{Q}_{mr}(W_B \mid W_A)} \cdot \frac{p(W_B \mid \mathbf{I})}{p(W_A \mid \mathbf{I})}\right). \tag{28}$$

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 (Schapire, 2000) as the number of features increases.

Figure 6(a) shows an approximation of the true posterior ratio by weights ω in a scope $\Omega_{mr}(W_A)$. In a continuous space, these particles shall be considered as centers for Parzen windows as a non-parametric representation for the space. In theory, we want to have this non-parametric form to cover the jump scope so that the Markov chain is ergodic. However, we also want to limit the window size so that the algorithm focuses more on the promising places. There needs to be further investigation on the theoretical analysis of this topic. In practice, we add a small variation to the particles when representing a continuous space.

We seek to enlarge the scope of jumps so that the Markov chain mixes rapidly with the ease of bottom-up proposals. This idea is illustrated in Figs. 6(b) and 6(c). With jumps of small scopes, the Markov chain needs more steps from a state W_A to a state W_B. If W_A and W_B are two distinct modes, this path will have a 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.

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:

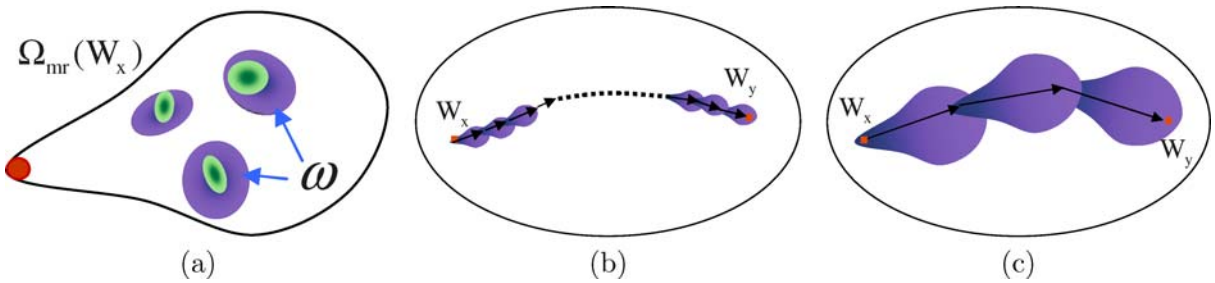


Figure 6. (a) Illustration of posterior probability p in scope $\Omega_{mr}(W_A)$ covered by the proposal probability $\mathbf{Q}_{mr}$. Darker points have high probabilities. (b). With small scopes of jumps there will be a long path or more steps between some states W_A and W_B. (b) The composite jumps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

(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 trees, (7) grouping/ungrouping parallel curves. (1–5) are simple jumps and (6–7) are composite jumps. The reversible jumps for regions are referred to a previous paper (Tu and Zhu, 2002).

4.1. *Bottom-Up Computation of the Candidate Set and Weights for Simple Jumps on Curves*

One key idea in the DDMCMC framework is to use discriminative models to compute the proposal probabilities. Some discussions about the interaction between discriminative and generative models are referred to Tu et al. (2003). This paper mostly focuse on the curve part.

In the discrete form, a curve C consists of a number of oriented bars in a chain. In a bottom-up process we compute an excessive number of candidates by a matching pursuit algorithm (Mallat and Zhang, 1993). An example is shown in Fig. (7). 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 with 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 which is discussed in Section 4.2.

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

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

Therefore we have a set of weighted atomic curves.

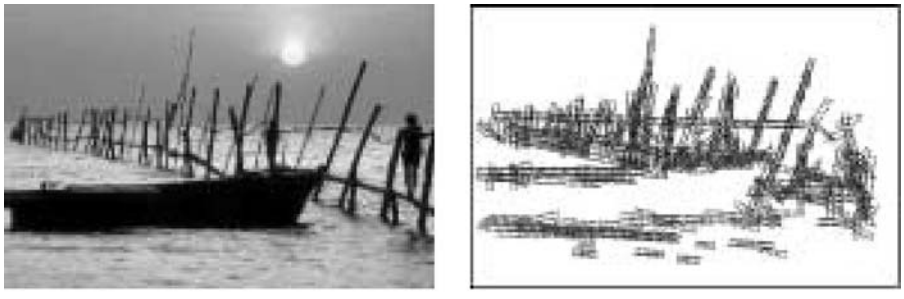
$$S_c = \left\{(c_i, \omega_i) : i = 1, 2, \ldots, M_c^{\mathrm{DD}}\right\}. \tag{29}$$

An example is shown in Fig. 7(b) where each atomic curve is represented by a bar. A new curve is 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 $p(\ell \mid F(\mathbf{I}))$ where $\ell \in \{+1, -1\}$ is the label for "curve" or "non-curve" respectively. $F(\mathbf{I})$ denotes a number of features in detection. By setting a low threshold in the ratio test $\frac{p(\ell=+1 \mid F(\mathbf{I}))}{p(\ell=-1 \mid F(\mathbf{I}))}$ 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 is attached to one end of an existing curve. 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



(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 bottom-up candidates.

two states W_A and W_B,

$$W_A = \left(W_-, K_A^c, \mathcal{PR}_A\right) \rightleftharpoons \left(W_-, c_B, K_B^c(c_B), \mathcal{PR}_B(c_B)\right) = W_B.$$

In the above notation, c_B is the new-born atomic curve, W_- denotes all other variables unchanged 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.

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

To design the reversible jumps, 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 it by $W = (W_-, c_{1r}, K_{1r}^c, \mathcal{PR}_+)$. $W = W_B$ 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 scope $\Omega_{mr}(W_A)$,

$$\mathbf{Q}_{1r}^*(W_B \mid W_A) = \frac{p(W_B \mid \mathbf{I})}{\sum_{W \in \Omega_{1r}(W_A)} p(W \mid \mathbf{I})} = \frac{\frac{p(W_B \mid \mathbf{I})}{p(W_A \mid \mathbf{I})}}{\sum_{c_{1r} \in S_{1r}} \frac{p(W \mid \mathbf{I})}{p(W_A \mid \mathbf{I})}}. \tag{30}$$

We divide both the nominator and the denominator by $p(W_A \mid \mathbf{I})$ since the probability ratios are much easier to compute due to cancellation. Note that the likelihoods $p(\mathbf{I} \mid W)$ and $p(\mathbf{I} \mid W_A)$ 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

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

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

$$\frac{p(W \mid \mathbf{I})}{p(W_A \mid \mathbf{I})} = \frac{p(\mathbf{I} \mid W)p(W)}{p(\mathbf{I} \mid W_A)p(W_A)} = \frac{p(\mathbf{I}_{\mathcal{D}(c_{1r})} \mid c_{1r})}{p(\mathbf{I}_{\mathcal{D}(c_{1r})} \mid R(c_{1r}))} \cdot p(c_{1r} \mid W_-) \cdot \frac{p(K^c(c_{1r}) \mid c_{1r})}{p(K_A^c)} \cdot \frac{p(\mathcal{PR}(c_{1r}) \mid c_{1r})}{p(\mathcal{PR}_A)}. \tag{31}$$

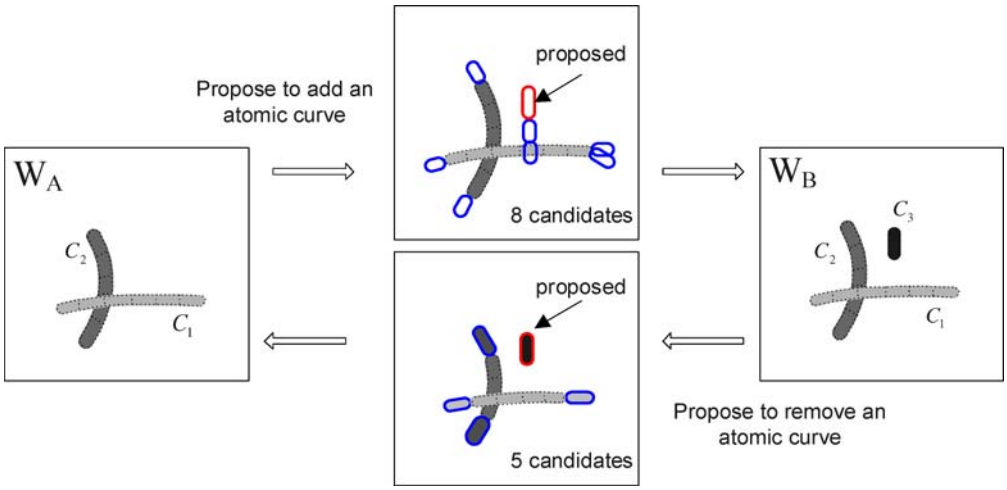


Figure 8. An example of the birth and death of a curve. At state W_A, there are 8 possible atomic curves to be proposed, which are shown as ellipses in the upper middle figure. W_B is the state after selecting an atomic curve to be a new curve. From W_B to W_A, there are 5 candidate atomic curves to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

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, and (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 are proposed by the context. They extend the existing curves.

For each candidate atomic curve $c_{1r}^{(i)}$, its weight $\hat{\omega_{1r}}^{(i)}$ approximates the factorized ratio in Eq. (31). 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, $p(\mathbf{I}_{\mathcal{D}(c_{1r})} \mid R(c_{1r}))$. This is available for the current W_A since we have computed each term in Eq. (15) for every existing regions and curves. (2). Its fitness to data $\mathbf{I}_{\mathcal{D}(c_{1r})}$ which is either computed for the data-driven candidates or from the context, $p(\mathbf{I}_{\mathcal{D}(c_{1r})} \mid c_{1r})$. 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, $\frac{p(\mathcal{PR}(c_{1r}) \mid c_{1r})}{p(\mathcal{PR}_A)}$. This is approximated by a uniform distribution. Thus, we have a set of weighted candidates for birth at W_A.

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

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

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

Similarly, we can design the death jump $\mathbf{J}_{1l}$ from W_B to W_A. Let $\Omega_{1l}(W_B)$ be the jump scope. We are interested in computing the probability for proposing W_A from $\Omega_{1l}(W_B)$. According to the MGS design, it is

$$\begin{aligned} \mathbf{Q}_{1l}^*(W_A \mid W_B) &= \frac{p(W_A \mid \mathbf{I})}{\sum_{W \in \Omega_{1l}(W_B)} p(W \mid \mathbf{I})} \\ &= \frac{\frac{p(W_A \mid \mathbf{I})}{p(W_B \mid \mathbf{I})}}{\sum_{c_{1l} \in S_{1l}} \frac{p(W \mid \mathbf{I})}{p(W_B \mid \mathbf{I})}} \end{aligned} \tag{32}$$

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

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

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,

$$\begin{aligned} \frac{p(W \mid \mathbf{I})}{p(W_B \mid \mathbf{I})} = \frac{p(\mathbf{I} \mid W)p(W)}{p(\mathbf{I} \mid W_B)p(W_B)} &= \frac{p(\mathbf{I}_{\mathcal{D}(c_{1l})} \mid R(c_{1l}))}{p(\mathbf{I}_{\mathcal{D}(c_{1l})} \mid c_{1l})} \\ &\times \frac{1}{p(c_{1l} \mid W_-)} \times \frac{p(K^c(c_{1l}) \mid c_{1l})}{p(K_B^c)} \\ &\times \frac{p(\mathcal{PR}(c_{1l}) \mid c_{1l})}{p(\mathcal{PR}_B)}. \end{aligned} \tag{33}$$

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,

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

The weight $\omega_{1l}^{(i)}$ is computed according to Eq. (33) where the factors have very intuitive meanings. $p(\mathbf{I}_{\mathcal{D}(c_{1l})} \mid R(c_{1l}))$ 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 a low chance to be removed. $p(\mathbf{I}_{\mathcal{D}(c_{1l})} \mid c_{1l})$ is available since it is computed in Eq. (15). $p(c_{1l} \mid W_-)$ is the prior of the atomic curve which is computed in Eq. (14).

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

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

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

$$\begin{aligned} \mathbf{K}_{1r}(W_A, W_B) &= Q_{1r}(W_B \mid W_A) \\ &\times \min\left(1, \frac{Q_{1l}(W_A \mid W_B)}{Q_{1\mathrm{r}}(W_B \mid W_A)} \cdot \frac{p(W_B \mid \mathbf{I})}{p(W_A \mid \mathbf{I})}\right), \end{aligned}$$

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

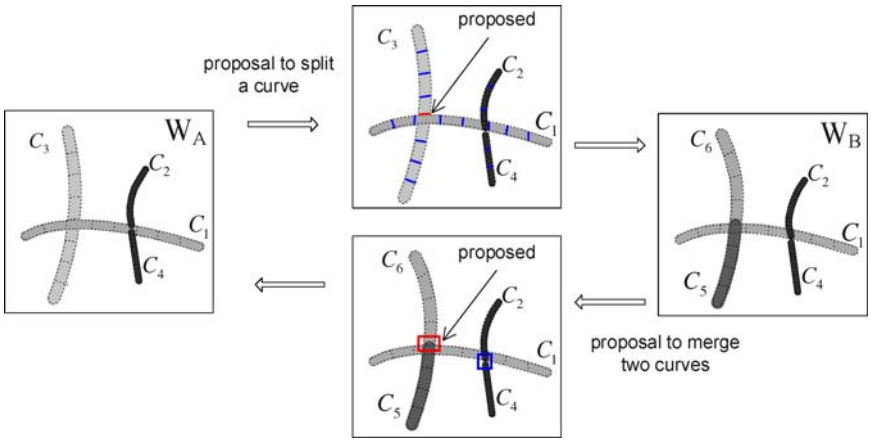


Figure 9. An example of the split-merge jumps for free curves. At state W_A, a set of 18 candidate sites are shown in the upper-middle figure. In this example, curve C_3 is split into two curves, C_5 and C_6. At W_B, there 2 candidate pairs which can be merged. They are shown in the lower-middle figure.

4.3. Jump II: Split and Merge of Curves

The second pair of reversible jumps $\mathbf{J}_2 = (\mathbf{J}_{2r}, \mathbf{J}_{2l})$ realize split-merge processes of free curves. Figure (9) shows an example. Similar to the birth-death jumps, we maintain two candidates lists at the current state.

$$S_{2r}(W_A) = \left\{ \left(z_{2r}^{(i)}, \omega_{2r}^{(i)}\right) : \ i = 1, 2, \ldots, N_{2r} \right\}, \text{and}$$
$$S_{2l}(W_B) = \left\{ \left(z_{2l}^{(i)}, \omega_{2l}^{(i)}\right) : i = 1, 2, \ldots, N_{2l} \right\}.$$

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. Figure (9) shows 18 candidate sites for split at W_A and 2 sites for merge at W_B.

The MGS proposal for split is,

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

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.

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

Again, we write the posterior ratio in a factorized form as

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

The four factors, again, have very intuitive meanings. (1) $\frac{p(\mathbf{I}_{\mathcal{D}(C_i) \cup \mathcal{D}(C_j)}) \mid C_i, C_j)}{p(\mathbf{I}_{\mathcal{D}(C_k)} \mid C_k)}$ measures the fitness for the curves before and after splitting. It probabilistically decides 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 more chance to be split. This is directly available in $p(\mathbf{I}_{\mathcal{D}(C_k)} \mid C_k)$ for the current state 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 $p(\mathbf{I}_{\mathcal{D}(C_i) \cup \mathcal{D}(C_j)}) \mid C_i, C_j)$ which is approximated by a similarity measure between their mean intensities. (2) $\frac{p(C_i)p(C_j)}{p(C_k)}$ reflects the priors only. If a curve is not so smooth, it should have high probability to be split. Intuitively, a 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 $p(\mathcal{PR}' \mid C_i, C_j)$ is approximated by a uniform distribu-

tion. The weight $\omega_{2r}^{(i)}$ will approximate the ratio $\frac{p(W \mid \mathbf{I})}{p(W_A \mid \mathbf{I})}$. The proposal probability is the weight normalized in the candidate set,

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

$\omega_{2r}^{(B)}$ 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 $\mathbf{Q}_{2r}(W_A \mid W_B)$.

4.4. *Jump III: Switching the Partial Order Relation*

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.

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

Each candidate $z_3^{(i)}$ is an occlusion between two existing curves $C_i \preceq C_j$ and a 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 partial order change. The weight of each candidate $z_3^{(i)}$ is only decided by the probability ratio on the overlapping image domain

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

All the junctions between free curves are collected in a candidate set Ξ, and they have equal probability. We compute the proposal probabilities for $\mathbf{J}_{3r}$ and $\mathbf{J}_{3l}$ the same way as for $\mathbf{J}_1$ 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 should switch to a curve. Conversely a short curve may become thick and switch to a region. This is realized by reversible jumps between two states,

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

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

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

The weights are decided by the priors for the curves and regions.

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

The proposal probabilities are computed as normalized weights within the two candidate sets. Since only priors about the curves and regions are involved, both $p(C_k)$ and $p(R_k)$ can be quickly computed at each step. Some simple tests such as the measurement of aspect ratio and area are adopted to make fast computation. 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.

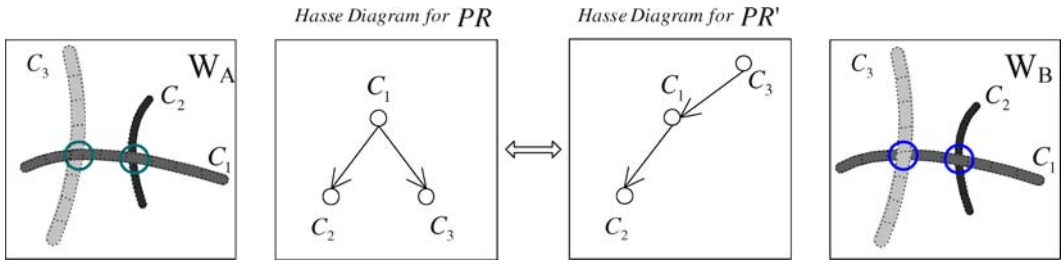


Figure 10. An example of partial order change.

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 (Tu and Zhu, 2002). Each type of model has a parameter space which has multiple modes. So the fifth pair of jumps $\mathbf{J}_5 = (\mathbf{J}_{5r}, \mathbf{J}_{5l})$ realize the switching 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 clustering methods, such as mean shift (Comaniciu and Meer, 1999) in the parameter space. Each candidate $(\ell^{(i)}, \theta^{(i)})$ is a mode in parameter space of type $\ell^{(i)}$.

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

When we switch intensity models for a region R or a curve C, the weight $\omega^{(i)}$ for the 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 fitness to the candidate model. Details about this part can be found in Tu and Zhu (2002).

4.7. Summary of the Five 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.

4.8. Jump VI: Split and Merge of Trees

The sixth pair of jumps are the split-merge of tree structures. They jump between two states,

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

Figure 11 illustrates an example where a tree T_k is split into trees T_i and T_j. 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.

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

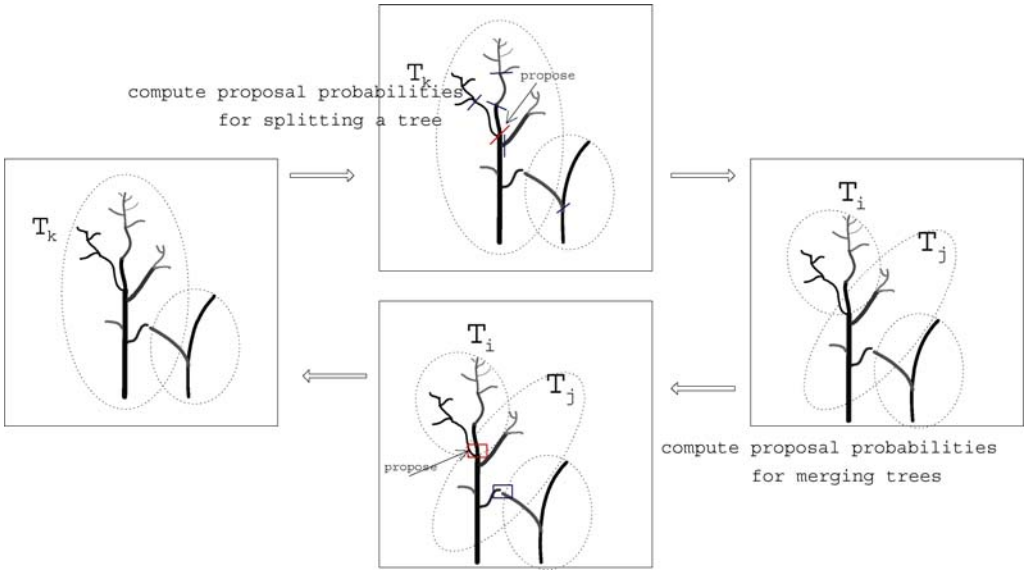


Figure 11. An example of splitting and merging trees.

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

The MGS proposal for splitting a tree is,

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

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.

$$\begin{aligned}\frac{p(W \mid \mathbf{I})}{p(W_A \mid \mathbf{I})} &= \frac{p(T_i)p(T_j)}{p(T_k)} \cdot \frac{p(K^T+1)}{p(K^T)} \\ &= \frac{1}{\exp\left\{-E_T\big(C_{T_k}(i), C_{T_k}(j)\big)\right\}} \cdot \frac{p(K^T+1)}{p(K^T)},\end{aligned}$$

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

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

$\omega_{6r}^{(B)}$ 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 $\exp\{-E_T(C_{\text{parent}}(z_B), C_{\text{child}}(z_B))\}$ at the site $z_{6r}^{(B)}$. The orders $\{\alpha_1, \alpha 2, \ldots, \alpha_n\}$ are decided by the tree structure in the parent-child relation directly.

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

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

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 $\{C_1, C_2, \ldots, C_N\}$ 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 $e_{st} =< C_s, C_t >$. Figure 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 (Barbu and Zhu, 2003, 2005) for partitioning this graph G. The SW-cut algorithm is a generalization to the Gibbs sampler.

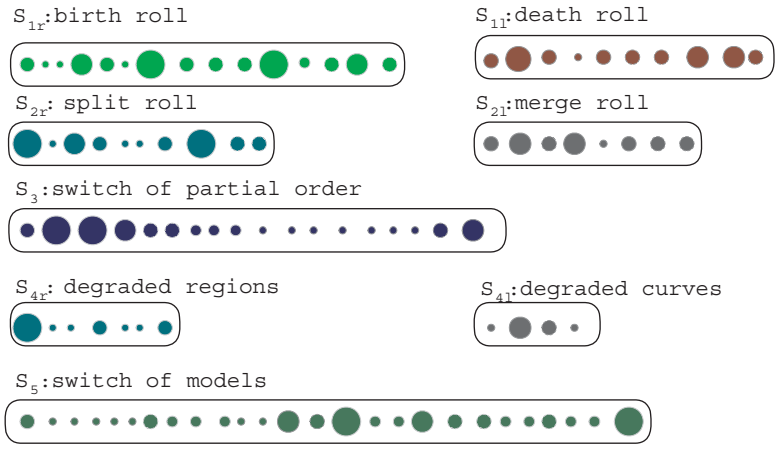


Figure 12. The 5 simple jumps maintain 10 sets of "particles" whose sizes illustrate their weights. The sets are updated and re-weighted in each jump steps, and they encode the proposal probabilities in a non-parametric representation.

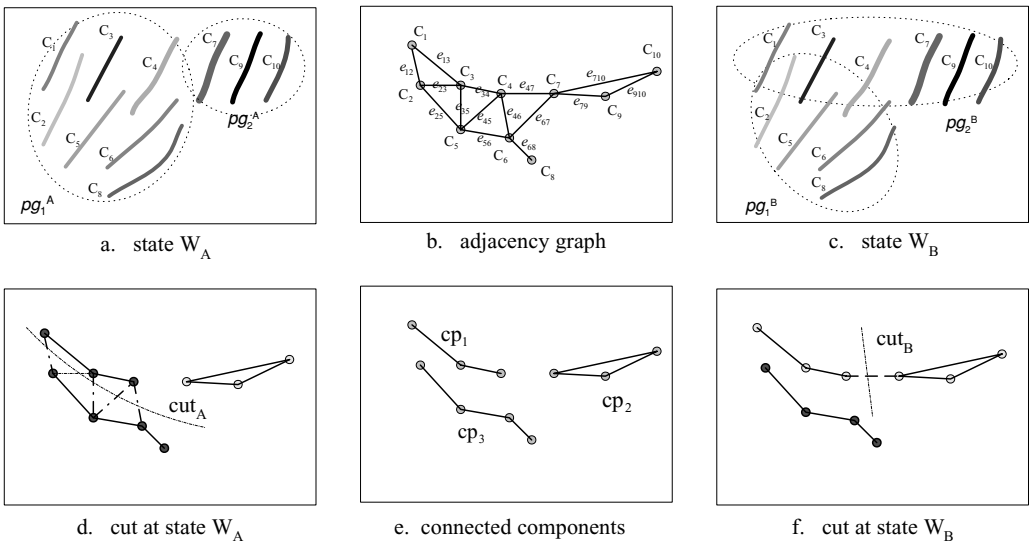


Figure 13. An example of split-merge of parallel curve groups by a composite jump designed with the SW-cut method.

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 Barbu and Zhu (2003) 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 Eq. (6).

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 $1 - q_{st}$.

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

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. 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 $\ell(CP_1) \in \{1, 2, 3, \ldots, K^{pg}\}$. We assign a new color $\ell'(CP_1) \in \{1, 2, \ldots, K^{pg} + 1\}$ to all curves in CP_1 at a proposal probability $q(\ell'(CP_1) \mid W)$.

According the the SW-cut algorithm(Barbu and Zhu, 2003, 2005), the proposal probability ratio is given by,

$$\frac{\mathbf{Q}(W_B \mid W_A)}{\mathbf{Q}(W_A \mid W_B)} = \frac{\prod_{e_{st} \in \text{Cut}_A}(1 - q_{st})}{\prod_{e_{st} \in \text{Cut}_B}(1 - q_{st})} \cdot \frac{q(\ell'(CP_1){=}2 \mid W_A)}{q(\ell'(CP_1){=}1 \mid W_B)}. \tag{34}$$

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,

$$\alpha(W_A, W_B) = \min\left(1, \frac{\mathbf{Q}(W_B \mid W_A)}{\mathbf{Q}(W_A \mid W_B)} \cdot \frac{p(pg_1^A)p(pg_2^A)}{p(pg_1^B)p(pg_2^B)}\right). \tag{35}$$

In the above equation, the posterior probability ratio $\frac{p(W_B \mid \mathbf{I})}{p(W_A \mid \mathbf{I}}$ 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.

5. Experiments

In Fig. 2 we have showed some examples where an image segmentation algorithm produces unsatisfactory results. This is 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 see 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.

The proposed algorithm searches for the optimal solution W^* by sampling $p(W \mid I)$. It starts from a segmentation with regions obtained at a coarse level by the Canny edge detector. Our method does not rely much on initial solution due to the use of various MCMC dynamics guided by bottom-up proposals, which help the algorithm to jump out of local minimums. However, we do use an annealing strategy to allow large change of W at high temperatures, and to focus more on local modes with the temperature gradually cooling down. The optimal solution W^* is found when the algorithm converges since $p(W^* \mid I)$ is in general highly peaked for many vision problems, especially at a low temperature. It is always desirable to avoid the use of annealing. As we discussed in Secton (3.3), this requires to design more efficient algorithms capable of making bigger scope of moves. Some of our recent attempts have made this possible for segmentation. In the case of maintaining multiple promising modes for the solution, we introduce a k-adventurers algorithm and the details can be found in Tu and Zhu (2002) .

Experiment A: Computing Regions and Free Curves.

Six examples are shown in Fig. (14) and (19). For each example, the first row 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 the synthesized image according to the generative models for the regions $\mathbf{I}^r_{\text{syn}} \sim p(\mathbf{I} \mid W^r)$, the curves $\mathbf{I}^c_{\text{syn}} \sim p(\mathbf{I} \mid W^c)$, and the overall synthesis $\mathbf{I}_{\text{syn}}$ by occluding $\mathbf{I}^c_{\text{syn}}$ on $\mathbf{I}^r_{\text{syn}}$.

We construct synthesis image to verify how an input image is represented in W^*. For example, Fig. (14) shows that the faces of the three Japanese ladies are treated as generic regions rather than high-level objects.

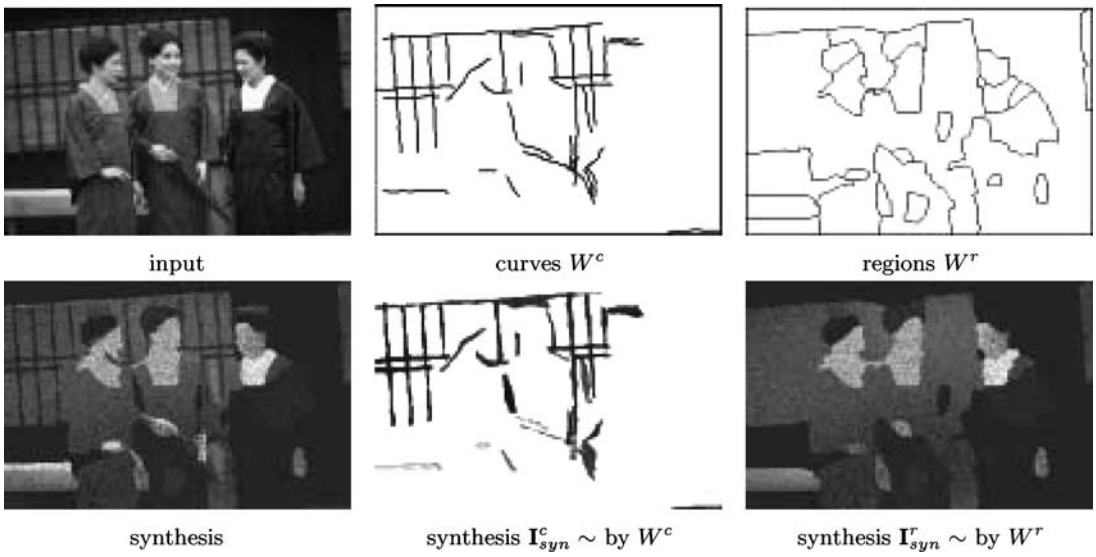


Figure 14. Experiment A1: parsing images into regions and free curves.

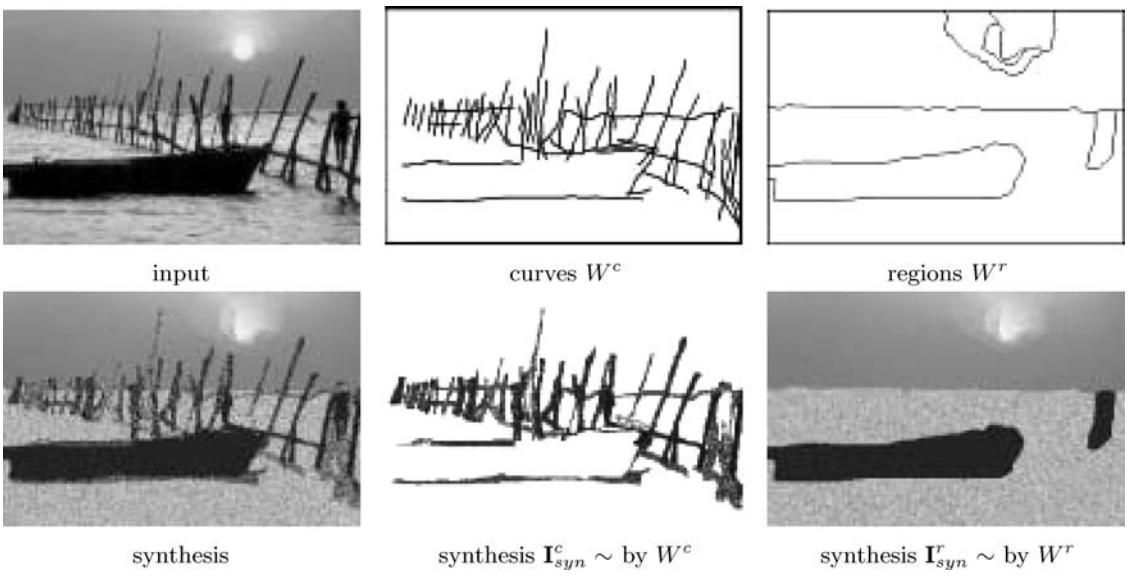


Figure 15. Experiment A2: parsing images into regions and free curves.

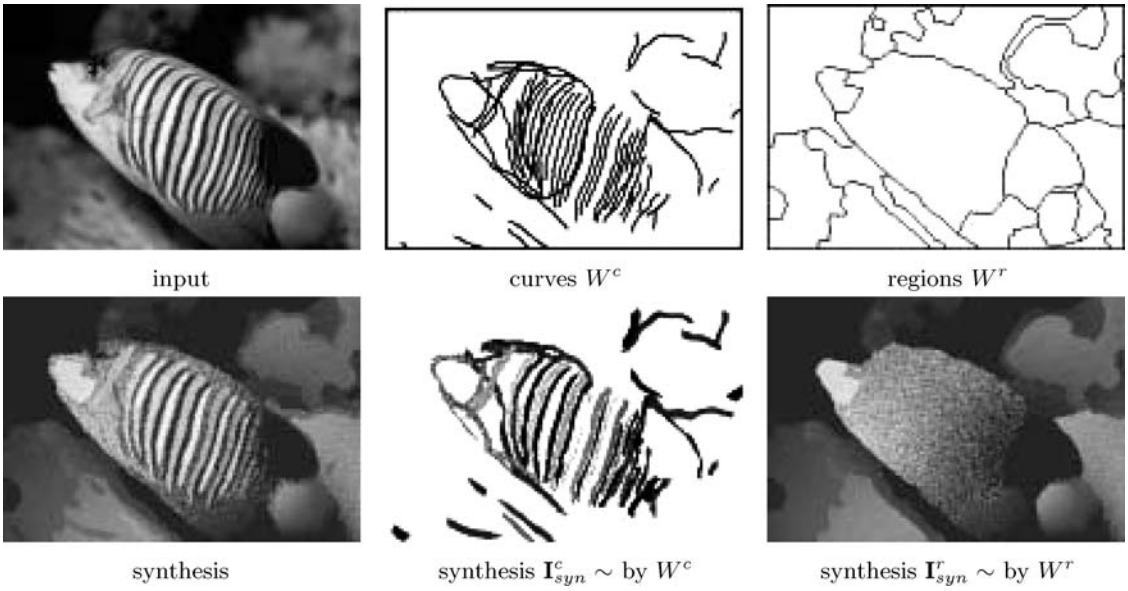


Figure 16. Experiment A3: parsing images into regions and free curves.

In these experiments, two parameters in the prior models are adjustable: (1) γ_r in equation (11), and (2) γ_c in Eq. (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)–(23) show four examples. In each example, the top row shows the parallel groups or trees grouped from the simple curves. The

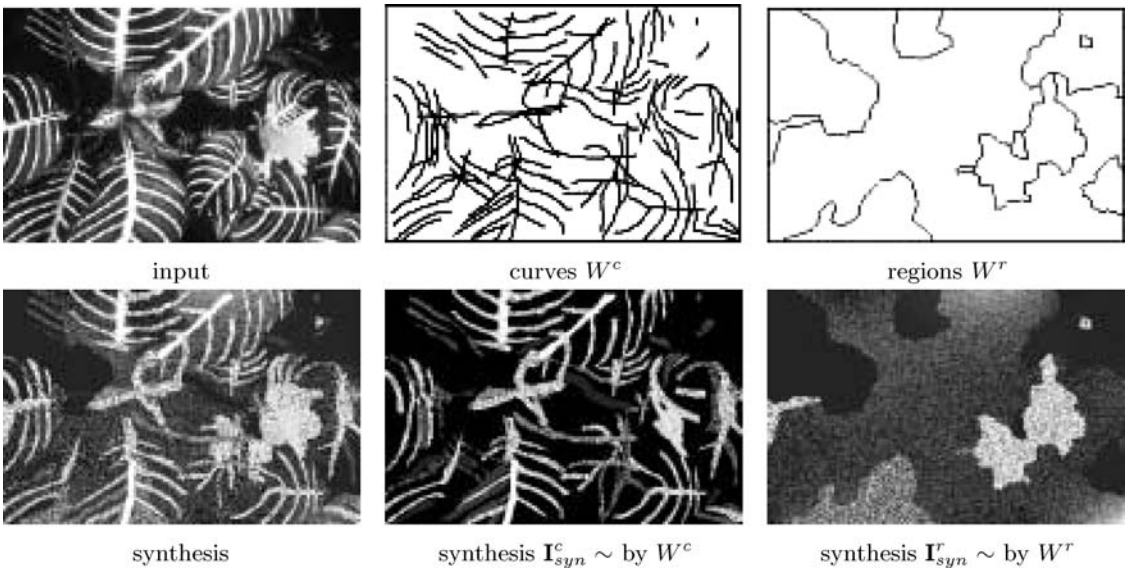


Figure 17. Experiment A4: parsing images into regions and free curves.

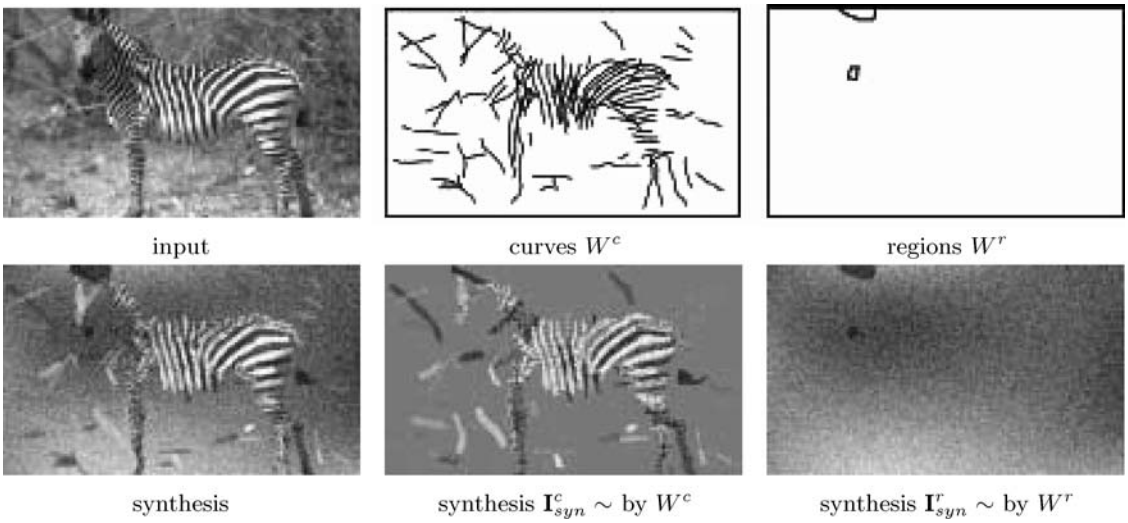


Figure 18. Experiment A5: parsing images into regions and free curves.

second and third rows are displayed as before. From the results, we can see that the algorithm successfully segments, detects, and groups regions, curves, and curve groups respectively.

We observe 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.

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 speed up the program in the future.

Validation of the Results. A benchmark dataset has been designed in the Berkeley vision lab Martin et al. (2001).

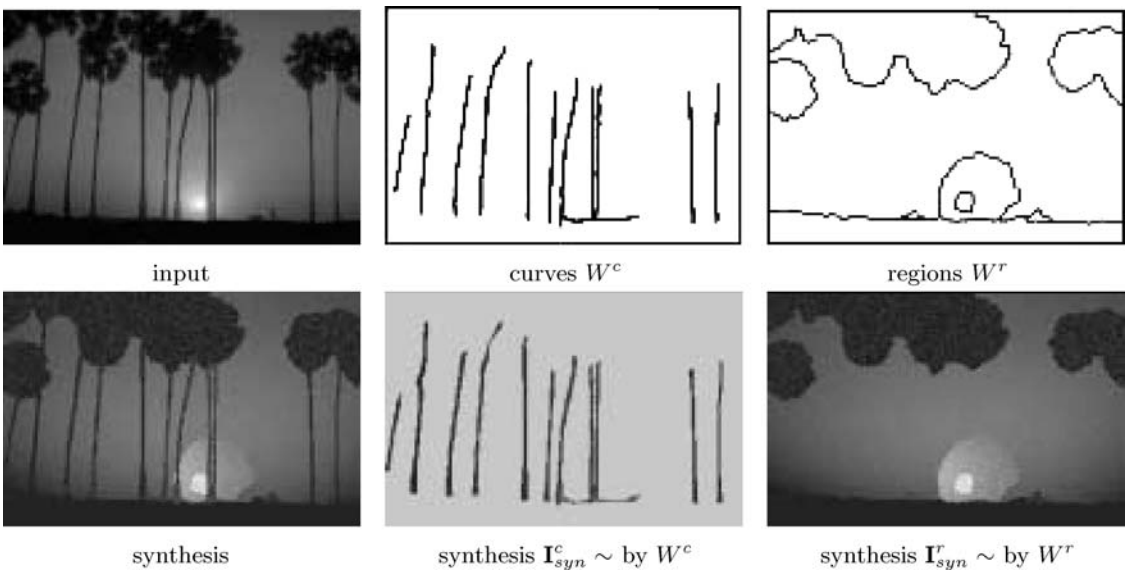


Figure 19. Experiment A6: parsing images into regions and free curves.

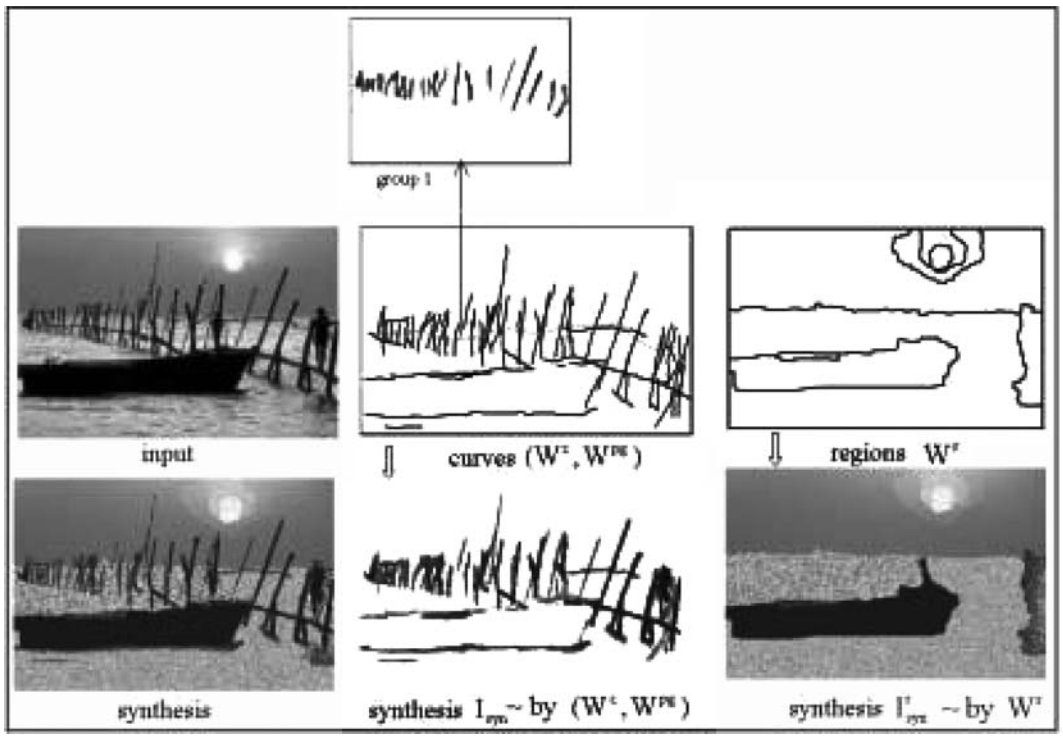


Figure 20. Experiment B1: parsing an image into regions, curves, and parallel curve groups.

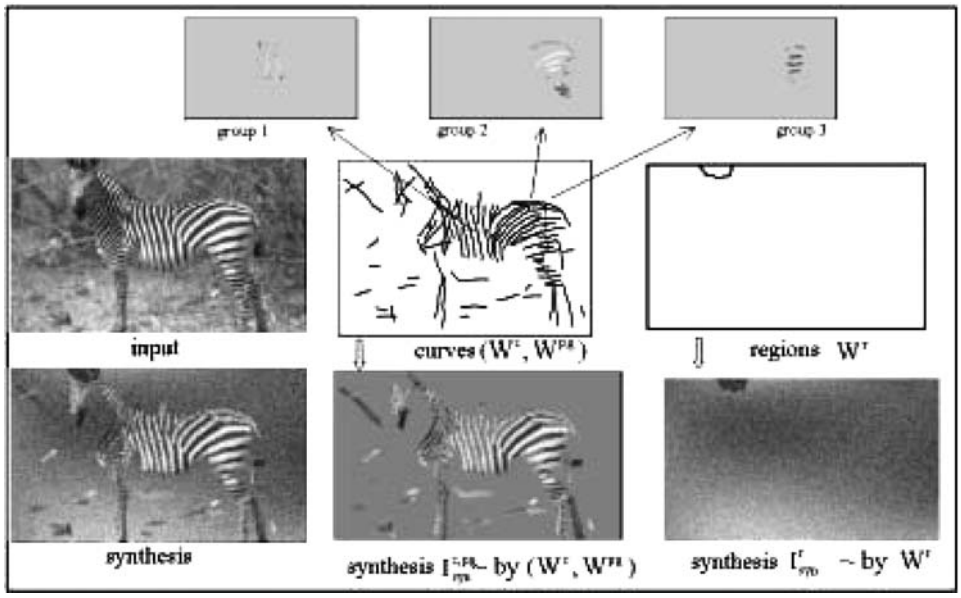


Figure 21. Experiment B2: parsing an image into regions, curves, and parallel curve groups.

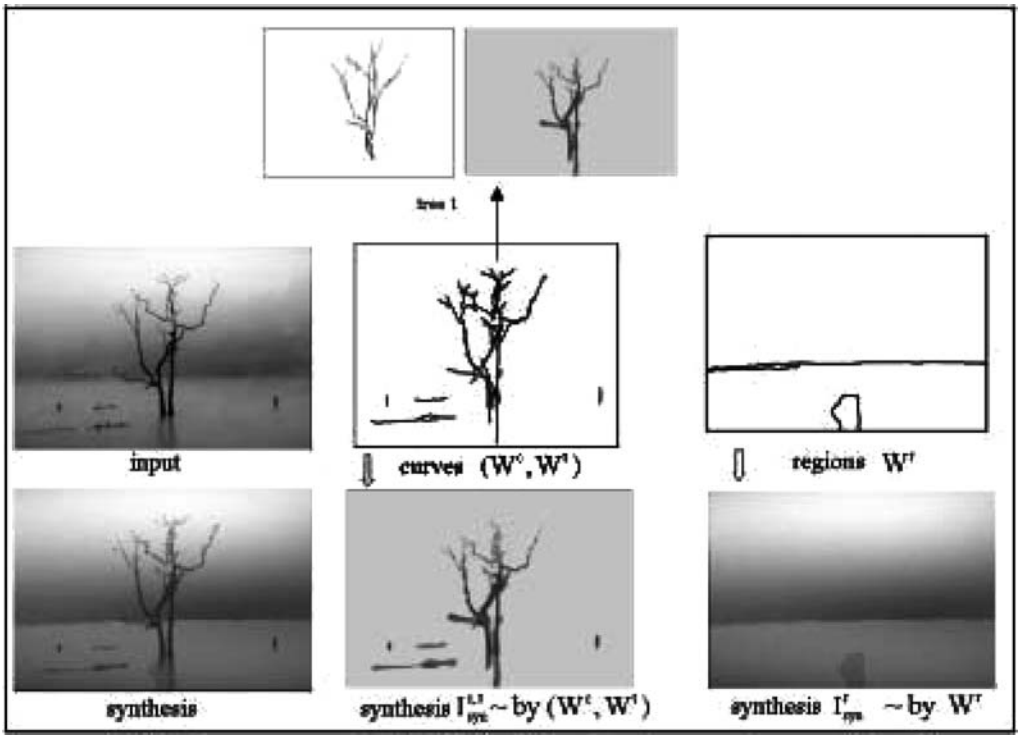


Figure 22. Experiment B3: parsing a tree image into regions, curves, and trees.

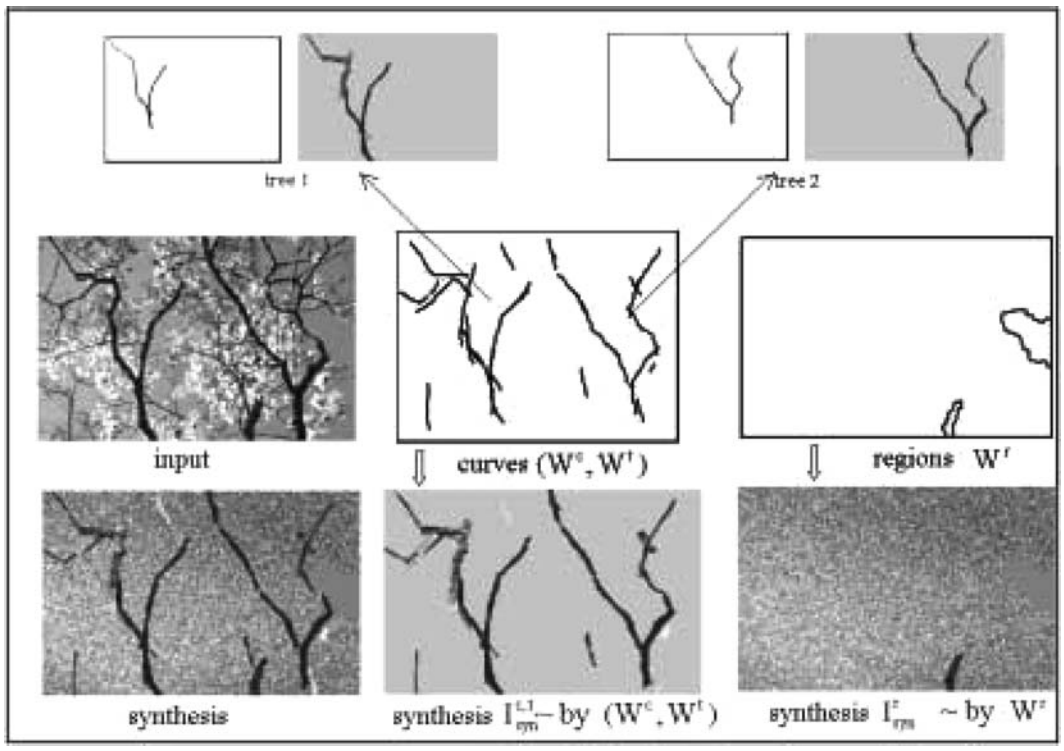


Figure 23. Experiment B4: parsing an image into regions, curves, and trees.

But this dataset is not emphasized on the segmentation of regions together with curves. We are putting some major efforts to build up a much larger dataset in which detailed structures are manually annotated for a variety of images. This will help us to validate our algorithms in the future.

6. Discussion and Future Work

In this paper, we introduce an 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 to compete to explain the image. The second aspect is the design of a 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.

The proposed algorithm improves segmentation results by explicitly modeling 1D curves, and degenerated regions. As a middle-level task, curve detection is useful in many vision problems such as tracking, object recognition, medical imaging, and structure from motion. Without any pre-assumption about the background, the method is able to automatically detect curves based on the integration of regions and curves. The representation computed in this work has been used in reconstructing 3D scene from a single image (Han and Zhu, 2003). This paper is part of a series of work integrating discriminative and generative models from low-level image segmentation (Tu and Zhu, 2002), middle-level curve structures detection (Tu and Zhu, 2002), to high-level object recognition (Tu et al., 2003).

However, there are many dynamics involved in the current framework. This poses difficulty in the implementation of the algorithm. It is due to two ma-

jor issues we need to further investigate in the future: (1) The jump dynamics are not general enough. Also, their scopes are still not big enough to encompass large structural changes. Therefore, our algorithm is yet not capable of quickly jumping among very distinct and promising modes. The SW-cut algorithm improves this aspect. But more research work still needs to be done along this vein. (2) The use of discriminative and generative models is quite separated. Our hope is to bring the bottom-up and top-down processes as close as possible in our future research. We are making progress along this line. There are some other modeling problems in the current framework. Though rich enough to represent many low-level and middle-level patterns, our generative models are still quite limited and it is a big challenge to study more complex high-level patterns. We need to bring more learning aspects into our framework to improve this part.

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

August, J. and Zucker, S.W. 2003. Sketches with curvature: The curve indicator random field and Markov processes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(4): 387–400.

Barbu, A. and Zhu, S.C. 2003. Graph partition by swendsen-wang cuts. In *Proc. Int'l Conf. Comp. Vis.*, Nice, France.

Barbu, A. and Zhu, S.C. 2005. Cluster sampling and its applications in image analysis. *Preprint 409* Department of Statistics, UCLA.

Bremaud, P. 1999. Markov chains: Gibbs fields, monte carlo simulation and queues. Springer. (chapter 6).

Bubley, R. and Dyer, M. 1997. Path coupling: A technique for proving rapid mixing in markov chains. In *Proc. of 38th Annual IEEE Symposium on Foundations of Computer Science*.

Candes, E.J. 1998. Ridgelets: Theory and applications. *Ph.D. Dissertation*, Yale University.

Carlsson, S. 1998. Sketch based coding of grey level images," *Signal Processing*, Vol. 15.

Comaniciu, D. and Meer, P. 1999. Mean shift analysis and applications. In *Proc. Int'l Conf. Comp. Vis*.

Cooper, C. and Frieze, A.M. 1999. Mixing properties of the swendsen-wang process on classes of graphs. *Random Structures and Algorithms*, Vol. 15, pp. 242–261.

Cootes, T.F., Cooper, D., Taylor, C.J., and Graham, J. 1995. Active shape models—Their training and application. *Computer Vision and Image Understanding*, 61(1):38–59.

Dellaert, F., Seitz, S., Thorpe, C., and Thrun, S. 2003. EM, MCMC, and Chain flipping for structure from motion with unknown correspondence. *Machine Learning*, No. 50, pp. 45–71.

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

Forsyth, D.A. 2001. The joy of sampling. *Int'l J. of Computer Vision*, 41(1/2):109–134.

Geman, S. and Geman, D. 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. PAMI*, 6:721–741.

Geman, D. and Jedynak, B. 1996. An active testing model for tracking roads from satellite images. *IEEE Trans. PAMI*, 18(1).

Parida, L., Geiger, D., and Hummel, R. 1998. Junctions: Detection, classification, and reconstruction. *IEEE Trans. PAMI*, 20(7).

Green, P.J. 1995. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732.

Grenander, U. and Miller, M.I. 1994. Representations of knowledge in complex systems. *J. of the Royal Stat. Soc. Series B*, 56(4):549–603.

Han, F. and Zhu, S.C. 2003. Bayesian reconstruction of 3d shapes and scenes from a single image. In *Proc. of Int'l Workshop on High Level Knowledge in 3D Modeling and Motion*, Nice, France.

Hastings, W.K. 1970. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109.

Isard, M. and Blake, A. 1996. Contour tracking by stochastic propagation of conditional density. In *Proc. of 4th European Conference on Computer Vision*.

Kaess, M., Zboinske, R., and Dellaert, F. 2004. Multiview reconstruction of piecewise smooth subdivision curves with a variable number of control points. *Proc. of 8th European Conference on Computer Vision*.

Khan, Z., Balch, T., and Dellaert, F. 2004. An MCMC-based particle filter for tracking multiple interacting targets. In *Proc. of 8th European Conference on Computer Vision*.

Kass, M., Witkin, A., and Terzopoulos, D. 1988. Snakes: Active contour models. *Int'l J. Computer Vision*, 1(4):321–332.

Lanterman, A.D. 2001. Jump-Diffusion algorithm for multiple target recognition using laser radar range data. *Optical Engineering*, 40(8):1724–1728.

Lee, M.W. and Cohen, I. 2004. Proposal maps driven MCMC for estimating human body pose. In *Proc. of CVPR*, Washington.

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

Maciuca, R. and Zhu, S.C. 2005. First-hitting-time analysis of independence metropolis sampler. *Journal of Theoretical Probability*(Accepted).

Malik, J., Belongie, S., Leung, T., and Shi, J. 2001. Contour and texture analysis for image segmentation. *Int'l J. Comp. Vis*.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. Int'l Conf. Computer Vision*, 2:416–423.

Mallat, S.G. and Zhang, Z. 1993. Matching pursuits with time-frequency dictionaries. *IEEE Tranc. on Signal Processing*, 41(12).

Metropolis, N., Rosenbluth, M.N., Rosenbluth, A.W., Teller, A.H., and Teller, E. 1953. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092.

Nitzberg, M. and Mumford, D.B. 1990. The 2.1D Sketch. In *Proc. Int'l Conf. Comp. Vis.*, pp.138–144.

Puskun, P.H. 1973. Optimum Monte Carlo sampling using Markov chains. *Biometrika*, 60(3):607–612.

Schapire, R.E. 2000. The boosting approach to machine learning: An overview. *MSRI Workshop on Nonlinear Estimation and Classification*.

Skiena, S. 1990. Partial orders. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Addison-Wesley: MA, pp. 203–209.

Shi, J. and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 22(8).

Srivastava, A, Grenander, U., Jensen, G.R., and Miller, M.I. 2002. Jump-diffusion Markov processes on orthogonal groups for object post estimation. *J. of Statistical Planning and Inference*, 103:15-37.

Swendsen, R.H. and Wang, J.S. 1987. Nonuniversal critical dynamics in monte carlo simulation. *Physical Review Letters*, 58(2):86–88.

Tu, Z. and Zhu, S.C. 2002. Image segmentation by data-driven markov chain monte carlo. *IEEE Trans. PAMI,* 24(5).

Tu, Z. and Zhu, S.C. 2002. Parsing images into region and curve processes. In *Proc. of 7th European Conference on Computer Vision*, Copenhagen.

Tu, Z., Chen, X., Yuille, A.L., and Zhu, S.C. 2003. Image parsing: Segmentation,detection and recognition. In *Proc. Int'l Conf. Comp. Vis.*, Nice, France.

Wang, J. and Adelson, E. 1994. Representing moving images with layers. *IEEE Trans. on Image Processing*, 6(3).

Zhao, T. and Nevita, R. 2004. Tracking multiple humans in crowded environment. In *Proc of CVPR*, Washington.

Zhu, S.C. and Yuille, A.L. 1996. Region competition: Unifying snakes, region growing, and bayes/mdl for multi-band image segmentation. In *IEEE Trans. PAMI*, 18(9):884–900.

Zhu, S.C., Zhang, R., and Tu, Z.W. 2000. Integrating top-down/bottom-up for object recognition by data driven markov chain monte carlo. In *Proc. of Int'l Conf. on Computer Vision and Pattern Recognition*, Hilton Head, SC.

Zimmer, C., Labruyere, E., Meas-Yedid, V., Guillen, N., Olivo-Marin, J.C. 2002. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours. *IEEE Trans. Medical Imaging*, 21(10).