

Range Image Segmentation by an Effective Jump-Diffusion Method

Feng Han, Zhuowen Tu, and Song-Chun Zhu

Abstract—This paper presents an effective jump-diffusion method for segmenting a range image and its associated reflectance image in the Bayesian framework. The algorithm works on complex real-world scenes (indoor and outdoor), which consist of an unknown number of objects (or surfaces) of various sizes and types, such as planes, conics, smooth surfaces, and cluttered objects (like trees and bushes). Formulated in the Bayesian framework, the posterior probability is distributed over a solution space with a countable number of subspaces of varying dimensions. The algorithm simulates Markov chains with both reversible jumps and stochastic diffusions to traverse the solution space. The reversible jumps realize the moves between subspaces of different dimensions, such as switching surface models and changing the number of objects. The stochastic Langevin equation realizes diffusions within each subspace. To achieve effective computation, the algorithm precomputes some importance proposal probabilities over multiple scales through Hough transforms, edge detection, and data clustering. The latter are used by the Markov chains for fast mixing. The algorithm is tested on 100 1D simulated data sets for performance analysis on both accuracy and speed. Then, the algorithm is applied to three data sets of range images under the same parameter setting. The results are satisfactory in comparison with manual segmentations.

Index Terms—Energy minimization, jump-diffusion, range segmentation, Markov chain Monte Carlo, data clustering, edge detection, Hough transform, change point detection.

1 INTRODUCTION

RESEARCH on range images in the past was mostly motivated by robotics applications for recognizing industry parts on assembly lines, and most of the work was focused on simple polyhedra objects. Recently, long-range laser scanners have become accessible to many users. These scanners can capture accurate 3D range images of real-world scenes at large scales. For example, Fig. 1 displays four panoramic range images of indoor and outdoor scenes scanned in cylindrical coordinates. Other examples are the 3D Lidar images in aerospace imaging which can capture depth maps of terrain and city scenes with accuracy up to 1 centimeter. As these range images provide much more accurate depth information than conventional vision cues, such as motion, shading, and binocular stereo, they are important for a wide variety of applications, for example, visualization of city scenes and historic sites [33], spatio-temporal databases of 3D urban and suburban developments, 3D scene modeling, and natural scene statistics analysis [22].

These new data sets pose much more serious challenges for range image analysis than the traditional polyhedra world. The main difficulties lie in the following aspects.

First, natural scenes contain many types of objects, for example, man-made objects (buildings, desks), animate objects (human and animals), and free-form objects (trees

and terrain). These objects should be represented by various families of surface models which have different parameterization dimensions. Formulated in the Bayesian framework, the posterior probability is distributed over a solution (state) space with a countable number of subspaces of varying dimensions. Each subspace is for a certain combination of surface models. Thus, an algorithm must accommodate multiple surface models in representation and explore the possible combinations of these models during computation.

Second, objects and surfaces in natural scenes appear at multiple scales. For example, the office scenes in Fig. 1 contain large surfaces such as walls, ceilings, and floors; middle size objects such as people, chairs and tables; and small objects such as books and cups on the desktop. This is in contrast with the block world (see Figs. 9 and 10) where objects are of similar sizes. In the presence of such a broad range of perceptual scales, it becomes difficult to segment small objects without oversegmenting large surfaces. Conventional model complexity criteria like MDL and AIC [34], [1] do not easily deal with such cases as they are based on image coding, not perception. Therefore, one should seek prior models to ensure proper segmentation of surfaces at all sizes.

Third, although range data is accurate in general, it is rather noisy around object boundaries. It becomes rather messy for objects like trees and bushes. Also, depth data are missing for objects at infinite distance like the sky and at metal, glass, and ceramic surfaces where the laser rays never return to the scanner.

In the literature, range image segmentation methods are mostly adapted from algorithms developed for optical images. For example, edge detection [24], region-based methods [2], [5], surface fitting [19], clustering [20], [12], and the generalized Hough transform [3] for detecting parametric surfaces of low dimension. We refer to Hoover et al. [21] for a

• The authors are with the Departments of Computer Science and Statistics, University of California, Los Angeles, 8130 Math Science Building, Box 951554, Los Angeles, CA 90095.

E-mail: hanf@cs.ucla.edu, {ztu, sczhu}@stat.ucla.edu.

Manuscript received 16 Dec. 2002; revised 14 Jan. 2004; accepted 26 Feb. 2004.

Recommended for acceptance by A. Rangarajan.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117967.

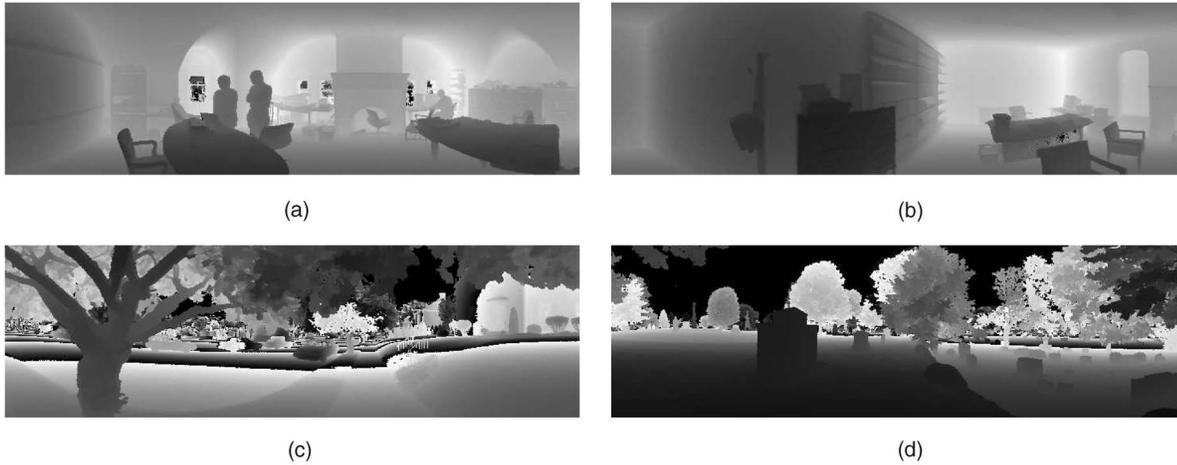


Fig. 1. Four examples of indoor and outdoor scenes from the Brown range data set. The laser scanner scans the scene in cylindrical coordinates and produces panoramic (wrapped) views of the scenes. (a) Office scene 1. (b) Office scene 2. (c) A street scene. (d) A cemetery scene.

survey of range segmentation algorithms and a good empirical comparison. Generally speaking, algorithms for range segmentation are not as advanced as those for intensity image segmentation. For example, there is no algorithm, to our knowledge, which can satisfactorily segment complex scenes such as those displayed in Fig. 1.

This paper presents a stochastic jump-diffusion algorithm for segmenting 3D scenes and reconstructing object surfaces from range images and their reflectance images. This work is an extension of our recent image segmentation work on data-driven Markov chain Monte Carlo [36]. It makes the following contributions to the range segmentation literature.

1. To deal with the variety of objects in real-world scenes, this algorithm incorporates five types of generative surface models, such as planes and conics for man-made objects, splines for free-form flexible objects, and a nonparametric (3D histogram) model for free-form objects (trees). These surfaces models compete to explain the range data under the regularity of a statistical prior for model complexity. The paper also introduces various prior models on surfaces, boundaries, and vertices (corners) to enforce regularities. The generative models and prior models are integrated in the Bayesian framework.
2. To handle missing range data, the algorithm uses the reflectance image as a different data channel and integrates it with the range data in the Bayesian framework. The reflectance image measures the proportion of laser energy returned from surface and therefore carries material properties. It is useful for segmenting glass, metal, ceramics, and the sky.
3. To explore the solution (state) space which consists of a countable number of subspaces of varying dimension, the algorithm simulates both reversible jumps and stochastic diffusions. The jumps realize split, merge, and model switching, while the diffusions realize boundary evolution and competition and model adaptation. These moves, in combination, simulate a Markov chain process sampling from the Bayesian posterior probability and the final solution is

obtained by annealing the temperature of the posterior probability.

4. To improve computational speed, the algorithm precomputes some bottom-up heuristics in a coarse-to-fine manner, such as edge detection and surface clustering at multiple scales. The computed heuristics are expressed as *importance proposal probabilities* [36] on the surfaces and boundaries which narrow the search space in a probabilistic fashion, and drive the Markov chain for fast mixing.

We first test the algorithm using 100 1D simulated range data sets for performance analysis where the ground truth is known. Then, we apply the algorithm to three data sets of range images. Two are the standard polyhedra data set and curved-surface data set from the vision group at University of South Florida, and the other data set, collected by the pattern theory group at Brown University, contains real-world scenes. The experiments demonstrate robust and satisfactory results under the same parameter setting and the results are compared with manual segmentation.

The paper is organized as follows: We present the jump-diffusion algorithm and evaluate its performance on 1D simulated data in Section 2. Then, we present a Bayesian formulation of the problem in Section 3 and the design of the algorithm in Section 4. Then, we show the experimental results in Section 5 and conclude the paper with some critical discussions in Section 6.

2 JUMP-DIFFUSION FOR 1D SIGNAL SEGMENTATION: A TOY EXAMPLE

This section presents the jump-diffusion algorithm for segmenting 100 simulated 1D range signals. With this simplified example, we focus on the fundamental ideas of the algorithm without becoming entangled in the complexity of real range images. Since the ground truth is known for the simulated data, we can evaluate how well the algorithm approaches the globally optimal solutions. We also compare the speeds of the algorithm with different designs and evaluate how much the various bottom-up heuristics expedite the Markov chain search.

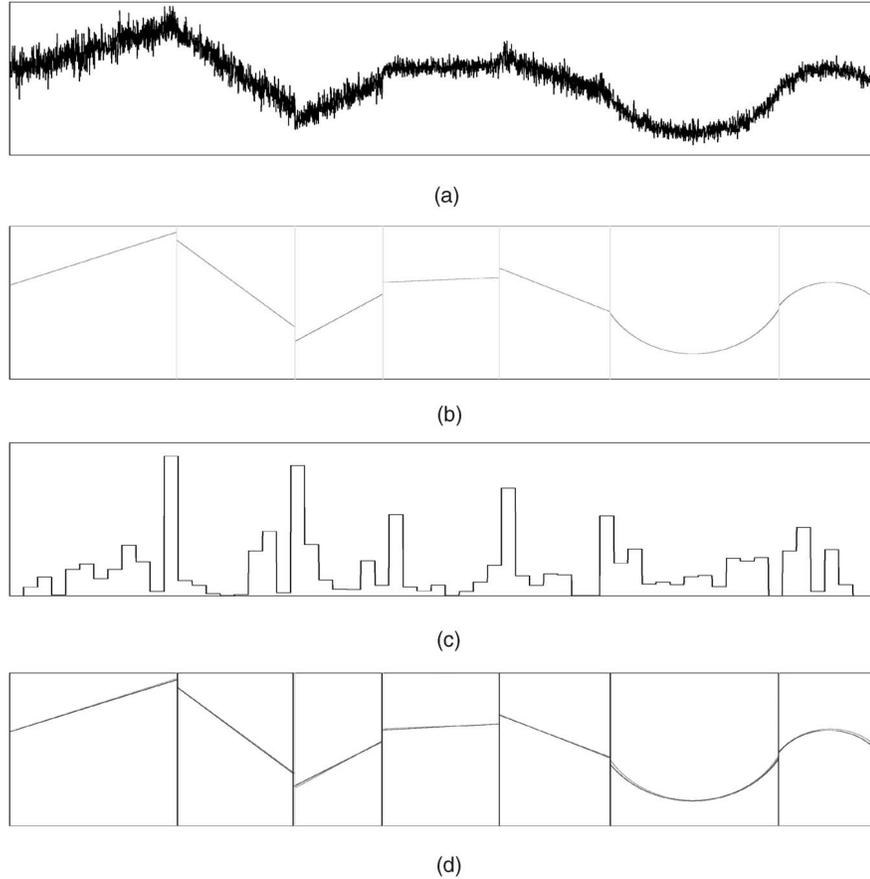


Fig. 2. (a) A 1D range image $\mathbf{I}(x)$, $x \in [0, 1]$. (b) The true segmentation W_{th} . (c) Edgerness measure $b(x)$, $x \in [0, 1]$. A large value $b(x)$ indicates a high probability for x being a change point. (d) The best solution W^* (dark gray) found by the algorithm plotted against W_{th} (light gray).

2.1 Segmenting 1D Range Data: Problem Formulation

Fig. 2a displays an example of a simulated 1D range image $\mathbf{I}(x)$, $x \in [0, 1]$. It is generated by adding Gaussian noise $N(0, \sigma^2)$ to the original surfaces \mathbf{I}_o in Fig. 2b. \mathbf{I}_o consists of an unknown number of k surfaces which could be either straight lines or circular arcs, separated by $k - 1$ change points,

$$0 = x_0 < x_1 < x_2 < \dots < x_{k-1} < x_k = 1.$$

Let $\ell_i \in \{\text{line}, \text{circle}\}$ index the surface type in interval $[x_{i-1}, x_i]$ with parameters θ_i , $i = 1, 2, \dots, k$. For a straight line, $\theta = (s, \rho)$ represents the slope s and intercept ρ . For a circular arc, $\theta = (\xi, \eta, \gamma)$ represents the center (ξ, η) and radius γ . Thus, the 1D "world scene" is represented by a vector of random variables,

$$W = (k, \{x_i : i = 1, 2, \dots, k-1\}, \{(\ell_i, \theta_i); i = 1, 2, \dots, k\}).$$

The surface \mathbf{I}_o is fully determined by W with $\mathbf{I}_o(x) = \mathbf{I}_o(x; \ell_i, \theta_i)$, $x \in [x_{i-1}, x_i]$, $i = 1, 2, \dots, k$.

By the standard Bayesian formulation, we have the posterior probability

$$p(W|\mathbf{I}) \propto \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^k \int_{x_{i-1}}^{x_i} (\mathbf{I}(x) - \mathbf{I}_o(x; \ell_i, \theta_i))^2 dx\right\} \cdot p(k) \prod_{i=1}^k p(\ell_i) p(\theta_i | \ell_i). \quad (1)$$

The first factor above is the likelihood and the rest are prior probabilities $p(k) \propto \exp^{-\lambda_0 k}$ and $p(\theta_i | \ell_i) \propto \exp^{-\lambda \# \theta_i}$, which penalize the number of parameters $\# \theta_i$. $p(\ell_i)$ is a uniform probability on the lines and arcs. Thus, an energy function is defined,

$$E(W) = \frac{1}{2\sigma^2} \sum_{i=1}^k \int_{x_{i-1}}^{x_i} (\mathbf{I}(x) - \mathbf{I}_o(x; \ell_i, \theta_i))^2 dx + \lambda_0 k + \lambda \sum_{i=1}^k \# \theta_i. \quad (2)$$

The problem is that W does not have a fixed dimension. The probability $p(W|\mathbf{I})$ (or the energy $E(W)$) is thus distributed over a countable number of subspaces of varying dimension. The following subsection briefly introduces the jump-diffusion process for exploring such solution spaces.

2.2 Background on Jump-Diffusion

In the statistics literature, there are some designs of *hybrid samplers* [35] which traverse parameter spaces of varying dimension by random choices of different Markov chain moves. Grenander and Miller [18] introduced the jump-diffusion process which mixed the Metropolis-Hastings method [29] and Langevin equations [14]. Other notable work includes [15] for reversible jumps and [31] for model comparison with reversible jumps. This section briefly

presents the basic ideas and discusses the main difficulty, namely, the computational speed.

In the 1D range segmentation problem above, let Ω denote the solution space which is a union of a countable number of subspaces

$$\Omega = \bigcup_{n=1}^{\infty} \Omega_n,$$

where $n = (k, \ell_1, \dots, \ell_k)$ indexes the subspaces. To traverse such solution space, the algorithm needs two types of moves: reversible jumps between different subspaces and stochastic diffusions within each continuous subspace.

1. **Reversible jumps.** Let $W = (n, \psi)$ be the state of a Markov chain at time t , where $\psi \in \Omega_n$ represents the continuous variables for the change points and the parameters of lines and arcs. In an infinitesimal time interval dt , the Markov chain jumps to a new space Ω_m ($m \neq n$) at state $W' = (m, \phi)$. There are three types of moves: 1) switching a line to a circular arc or vice versa, 2) merging two adjacent intervals to a line or a circle, and 3) splitting an interval into two intervals (lines or circles).

The jump is realized by a Metropolis move [29] which proposes to move from (n, ψ) to (m, ϕ) ($m \neq n$) by a forward proposal probability $q(n \rightarrow m)q(\phi|m)d\phi$. The backward proposal probability is $q(m \rightarrow n)q(\psi|n)d\psi$. The forward proposal is accepted with probability

$$\alpha((n, \psi) \rightarrow (m, \phi)) = \min\left(1, \frac{q(m \rightarrow n)q(\psi|n)d\psi \cdot p(m, \phi|\mathbf{I})d\phi}{q(n \rightarrow m)q(\phi|m)d\phi \cdot p(n, \psi|\mathbf{I})d\psi}\right). \quad (3)$$

The dimension is matched in the above probability ratio.

2. **Stochastic diffusions.** Within each subspace Ω_n with $n = (k, \ell_1, \dots, \ell_k)$ fixed, the energy functional $E(W)$ is

$$\begin{aligned} E[\psi] &= E(x_1, \dots, x_{k-1}, \theta_1, \dots, \theta_k) \\ &= \frac{1}{2\sigma^2} \sum_{i=1}^k \int_{x_{i-1}}^{x_i} (\mathbf{I}(x) - \mathbf{I}_o(x; \ell_i, \theta_i))^2 dx + \text{const.} \end{aligned}$$

We adopt a stochastic diffusion (or Langevin) equation to explore the subspace. The Langevin equations are the steepest descent PDE (partial differential equations) driven by Brownian motions $dB(t)$ with temperature T . Let $\psi(t)$ denote the variables at time t ,

$$d\psi(t) = -\frac{dE[\psi]}{d\psi} dt + \sqrt{2T(t)} dw_t, \quad dw_t \sim N(0, (dt)^2). \quad (4)$$

For example, the motion equation of a change point x_i is

$$\begin{aligned} \frac{dx_i(t)}{dt} &= \frac{1}{2\sigma^2} ((\mathbf{I}(x) - \mathbf{I}_o(x; \ell_{i-1}, \theta_{i-1}))^2 \\ &\quad - (\mathbf{I}(x) - \mathbf{I}_o(x; \ell_i, \theta_i))^2) + \sqrt{2T(t)} N(0, 1). \end{aligned}$$

This is the 1D version of the region competition equation [39]. The movement of the point x_i is driven by the fitness of data $\mathbf{I}(x_i)$ to the surface

models of the two adjacent intervals plus a Brownian motion. In practice, the Brownian motion is found to be useful in avoiding local pitfalls.

For computing the parameters $\theta_i, i = 1, 2, \dots, k$, running the diffusion is more robust and often faster than fitting the best θ_i for each interval $[x_{i-1}, x_i]$ deterministically since the deterministic fit is an ‘‘overcommitment.’’ It is especially true when the current interval contains more than one objects.

It is well-known [14] that the continuous Langevin equations in (4) simulate Markov chains with stationary density

$$p(\psi) \propto \exp\left\{-\frac{E(\psi)}{T}\right\}.$$

This is the posterior probability within subspace Ω_n at temperature T .

3. **The coordination of jumps and diffusions.** The continuous diffusions are interrupted by jumps at time instances $t_1 < t_2 < \dots < t_M \dots$ as Poisson events. In practice, the diffusion always runs at a discrete time step Δt . Thus, the discrete waiting time τ_j between two consecutive jumps is

$$w = \frac{t_{j+1} - t_j}{\Delta t} \sim p(w) = e^{-\tau} \frac{\tau^w}{w!},$$

where the expected waiting time $E[w] = \tau$ controls the frequency of jumps. Both jump and diffusion processes should follow an annealing scheme for lowering the temperature gradually.

For illustration, Fig. 4a shows two trials (thin and thick curves respectively) of the jump-diffusion process running on the input 1D range data in Fig. 2a. The energy plots go up and down (i.e., the algorithm is not greedy) and the continuous energy curves (diffusion) are interrupted by jumps.

4. **Reversibility and global optimization.** From an engineering point of view, the most important property of the jump-diffusion process is that it simulates Markov chain to traverse the complex solution space. This property distinguishes it from greedy and local methods. In theory, this Markov chain samples from the posterior probability $p(W|\mathbf{I})$ over the solution space Ω [18]. With an annealing scheme, it can theoretically achieve the globally optimal solution with probability close to one. The reversibility of the jumps may not be a necessary condition; however, it is a useful tool for achieving irreducibility of the Markov chain in the complex solution space. To the best of our knowledge, there is no alternative way for traversing such complex spaces except exhaustive searches.
5. **The speed bottlenecks.** Conventional jump-diffusion designs are limited by their computing speed. However, this problem can be overcome by better design of the proposal probabilities as we shall show in the next section. We observed that the bottlenecks are in the jumps affected by the design of the proposal probabilities. In (3), a proposal probability $q(\phi|m)$ in

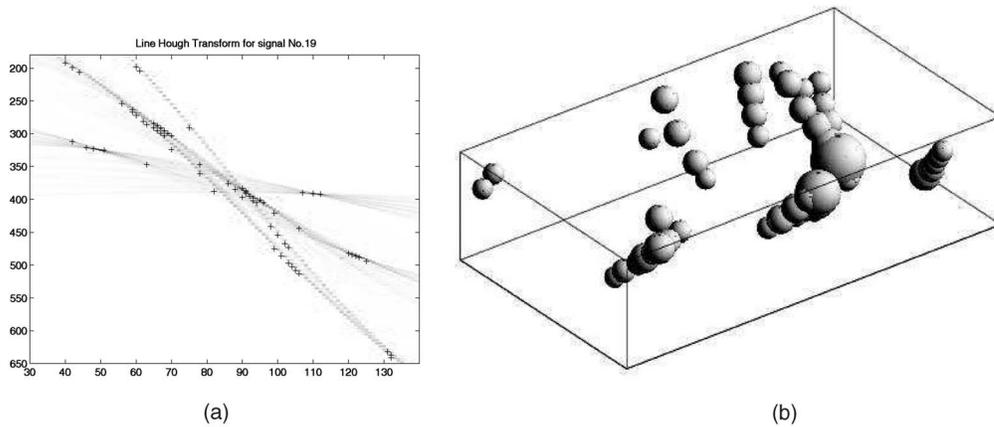


Fig. 3. Results of Hough transforms on the signal in Fig. 2a (a) Hough transform for lines. The line model space (s, ρ) . (b) Hough transform for arcs. The circle model space (ξ, η, γ) .

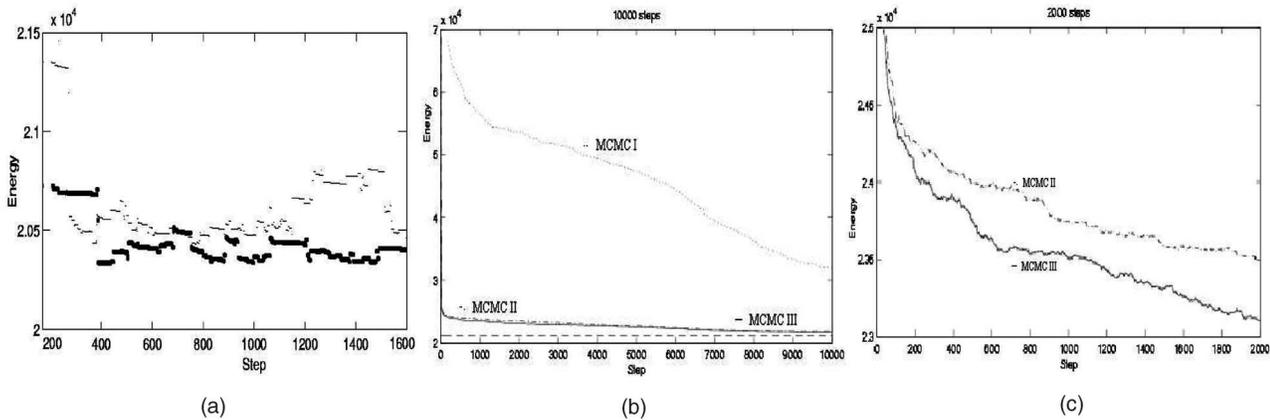


Fig. 4. (a) Diffusion with jumps. Energy plots for two trials (MCMC II—thin curve and MCMC III—thick curve) of the jump-diffusion processes. Continuous energy changes in diffusion are interrupted by energy jumps. (b) Average energy plot. Comparison of the energy curves in the first 10,000 steps of three Markov chains MCMC I, II, and III averaged over 100 randomly generated signals. (c) Zoomed-in view of MCMC II and III for the first 2,000 steps. Note the energy scale is different from (b).

interval $[x_{i-1}, x_i)$ can be divided into three cases: 1) switching to a new model with $\phi = \theta_i$, 2) merging to form a new interval $[x_{i-2}, x_i)$ with type ℓ and parameter ϕ , and 3) splitting to form two new intervals with models (ℓ_a, θ_a) and (ℓ_b, θ_b) , respectively.

$$q(\phi|m) = \begin{cases} q(\theta_i|\ell_i, [x_{i-1}, x_i)) & \text{switch } [x_{i-1}, x_i) \text{ to model } (\ell_i, \theta_i); \\ q(\theta|\ell, [x_{i-2}, x_i)) & \text{merge to a model } (\ell, \theta); \\ q(x|[x_{i-1}, x_i))q(\theta_a|\ell_a, [x_{i-1}, x)) \\ \quad q(\theta_b|\ell_b, [x, x_i)) & \text{split } [x_{i-1}, x_i) \text{ into } (\ell_a, \theta_a) \text{ and } (\ell_b, \theta_b) \text{ at } x. \end{cases} \quad (5)$$

In the statistical literature [18], [15], the proposal probabilities were taken mostly as uniform distributions, i.e., jumps to randomly selected lines or/and circles for new models. Such proposals are almost always rejected because the ratio $p(m, \phi|\mathbf{I})/p(n, \psi|\mathbf{I}) = e^{-\Delta E}$ could be close to zero. This causes extremely long waiting. We can overcome this problem by a better design than the uniform proposal probabilities, as the data-driven Markov chain Monte Carlo method has shown [36].

2.3 Data-Driven Techniques for Design Proposal Probabilities

The basic concept of a data-driven Markov chain Monte Carlo scheme [36] is to compute proposal probabilities by bottom-up methods in each of the parameter spaces for x , (*line*, s, ρ), and (*arc*, ξ, η, γ), respectively. In the 1D range signal, we use Hough transformation and change-point detection.

1. **Hough transformation in the model spaces.** Fig. 3a shows the Hough transform [3] in the line space (i.e., plane $\theta = (s, \rho)$). The crosses are detected as line candidates $\theta_{\text{line}}^{(1)}, \theta_{\text{line}}^{(2)}, \dots, \theta_{\text{line}}^{(N_{\text{line}})}$. Fig. 3b is the Hough transform result on the circular arc space $\theta = (\xi, \eta, \gamma)$ with bounds. The balls are arc candidates $\theta_{\text{arc}}^{(1)}, \theta_{\text{arc}}^{(2)}, \dots, \theta_{\text{arc}}^{(N_{\text{arc}})}$ with the sizes representing the weights (total number of votes received). Thus, when we propose a new model for an interval $[a, b)$, we compute the *importance proposal probability* by Parzen windows centered at the candidates.

$$q(\theta|\ell, [a, b)) = \sum_{i=1}^{N_\ell} \omega_i G(\theta - \theta_\ell^{(i)}), \quad \ell \in \{\text{line}, \text{arc}\}. \quad (6)$$

ω_i is the accumulated weights voted from the data in $[a, b)$ and $G(x)$ is a Parzen window centered at 0.

2. *Change point detection in the x space.* Fig. 2c shows the result of an edge strength $f(x|\nabla G * \mathbf{I}, \nabla^2 G * \mathbf{I})$ based on two filters: the first and second derivatives of a Gaussian shape. Instead of making a hard decision which is bound to be unreliable, we treat the strength measure as a probability. Thus, the proposal for a change point x is

$$q(x | [a, b)) = \frac{f(x|\nabla G * \mathbf{I}, \nabla^2 G * \mathbf{I})}{\int_a^b f(x|\nabla G * \mathbf{I}, \nabla^2 G * \mathbf{I}) dx}. \quad (7)$$

2.4 Speed Analysis and Empirical Evaluation

Replacing the uniform proposal probabilities by data-driven probabilities does not degrade the jump-diffusion process to a local algorithm. First, the data-driven proposal probabilities in (6) and (7) are nonzero in the entire parameter spaces, so the moves are still reversible and the Markov chain still can traverse the space. Second, by lowering the threshold, the chance of missing a line or arc in the Hough transforms can be made zero because any line or arc with nontrivial length should receive a number of votes in proportion to its length. The proposal probabilities only narrow the focus of the Markov chain search probabilistically.

Recent theoretical analysis [27] confirms our observation. Consider a finite space with M (very large) states $\Omega_o = \{1, 2, \dots, M\}$ and a posterior probability $p = (p_1, \dots, p_M)$. We sample p by an independent Metropolis method with proposal probability $q = (q_1, \dots, q_M)$.

Theorem Maciucă and Zhu [27]. *For the independent Metropolis sampler above, let $\tau(i)$ be the first hitting-time when the Markov chain first visits a state $i \in \Omega_o$ and $E[\tau(i)]$ its expectation. Then,*

$$\frac{1}{\min\{p_i, q_i\}} \leq E[\tau(i)] \leq \frac{1}{\min\{p_i, q_i\}} \cdot \frac{1}{1 - \|p - q\|}, \quad \forall i \in \Omega_o, \quad (8)$$

where $\|p - q\| = \frac{1}{2} \sum_{i \in \Omega_o} |p_i - q_i|$ measures the difference between q and p .

The theorem indicates that, if the bottom-up proposal q is close to p , the Markov chain will hit the optimal state in about $1/p^*$ steps on average with $p^* = \max\{p_1, \dots, p_M\}$. For posterior probabilities p with low entropy, this means a small number of steps even though the space size M is huge.

However, for a more complex space Ω , we have not yet been able to analytically link the design of $q()$'s to the convergence rate. Thus, we seek empirical comparison on the performance of various designs.

A set of 100 1D range data sets (like Fig. 2a) are simulated randomly with the known true segmentation. Three Markov chain designs are compared over the 100 1D range data.

MCMC I uses uniform distributions for $q()$ s without data-driven heuristics. MCMC II uses Hough transform results for $q(\theta | \ell, [a, b))$ and a uniform distribution for $q(x | [a, b))$, i.e., the change point detection heuristics are omitted. MCMC III uses both the Hough transform and change point detection heuristics for proposals.

Fig. 2d displays the optimal solution W^* found by MCMC III which coincides with the ground truth with a relative energy error under 0.2 percent. Fig. 4a shows the energy $E(W)$ against running time for the input in Fig. 2a by the thin curves (MCMC II) and thick curves (MCMC III). We select a number of time steps to only show the jump and diffusion processes in this figure.

The main results are shown in Figs. 4b and 4c. The Markov chains all start with randomized initialization and follow annealing schemes carefully tuned to the three cases. We plot the energy changes averaged over 100 signals for 10,000 steps. The energy jumps disappear because of averaging. In Fig. 4b, the dotted curve is for MCMC I, the dash-dotted curve is for MCMC II, and the solid curve is for MCMC III. The bottom line is the average "true" global optimal energy. Clearly, the MCMC II and MCMC III always achieve solutions near the ground truth (with relative energy error under 2 percent in the first 10,000 steps). The bottom-up heuristics drastically improve the speed in comparison with MCMC I. Fig. 4c is a zoomed-in view of the first 2,000 steps of MCMC II and MCMC III. It shows that change-point detection heuristics offer only a little bit of improvement.

3 BAYESIAN FORMULATION: INTEGRATING CUES, MODELS, AND PRIOR

This section formulates the problem of 2D range segmentation and surface reconstruction under the Bayesian framework by integrating two cues, five families of surface models, and various prior models.

3.1 Problem Formulation

We denote an image lattice by $\Lambda = \{(i, j) : 0 \leq i \leq L_1, 0 \leq j \leq L_2\}$. A range scanner captures two images. One is the 3D range data which is a mapping from lattice Λ to a 3D point,

$$\mathbf{D} : \Lambda \rightarrow \mathcal{R}^3, \quad \mathbf{D}(i, j) = (x(i, j), y(i, j), z(i, j)).$$

(i, j) indexes a laser ray that hits a surface point (x, y, z) and returns. The other image is a reflectance map

$$\mathbf{I} : \Lambda \rightarrow \{0, 1, \dots, G\}.$$

$\mathbf{I}(i, j)$ is the portion of laser energy returned from point $\mathbf{D}(i, j)$. $\mathbf{I}(i, j)$ is small at surfaces with high specularity, such as glass, ceramics, and metals that appear dark in \mathbf{I} , and $\mathbf{I}(i, j) = 0$ for mirrors and the sky. $\mathbf{D}(i, j)$ is generally very noisy and unreliable when $\mathbf{I}(i, j)$ is low, and is considered missing if $\mathbf{I}(i, j) = 0$.

The objective is to partition the image lattice into an unknown number of K disjoint regions,

$$\Lambda = \cup_{n=1}^K R_n, \quad R_n \cap R_m = \emptyset \quad \forall m \neq n.$$

Since natural scenes contain different types of surfaces like the 1D example, at each region R the range data is fit to a surface model of type ℓ^D with parameter Θ^D and the reflectance is fit to a reflectance model of type ℓ^I with parameter Θ^I . We denote a solution by

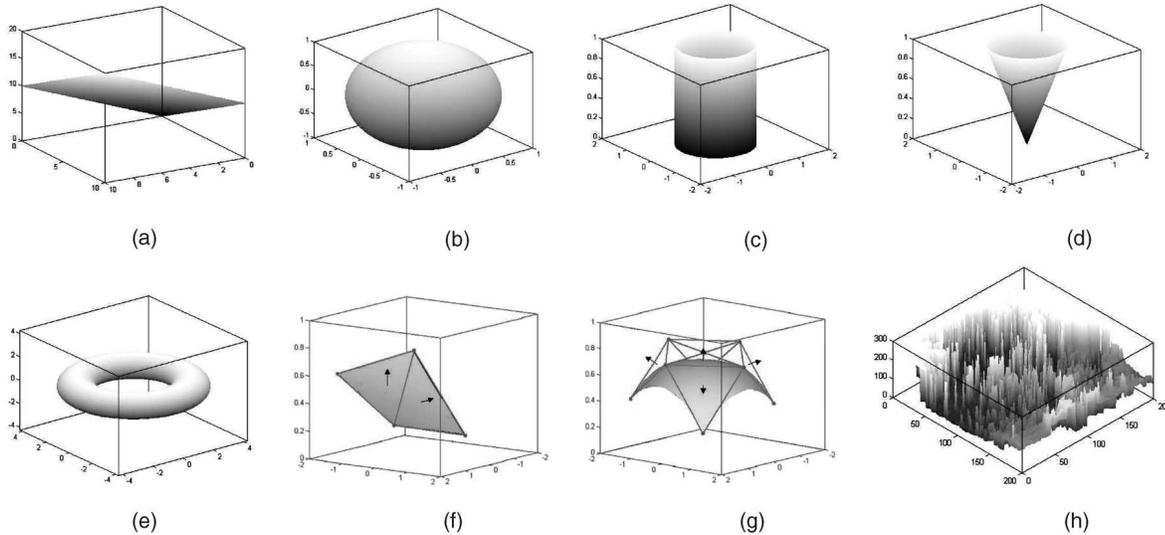


Fig. 5. Some typical surfaces for the five families of surfaces for fitting the range data D . (a) Plane. (b) Sphere/ellipsoid. (c) Cylinder. (d) Cone. (e) Torus. (f) 4-point spline. (g) 9-point spline. (h) Clutter.

$$W = (K, \{R_i : i = 1, 2, \dots, K\}, \{(\ell_i^D, \Theta_i^D), (\ell_i^I, \Theta_i^I) : i = 1, 2, \dots, K\}).$$

The algorithm should maximize a Bayesian posterior probability over a solution space $\Omega \ni W$,

$$W^* = \arg \max_{W \in \Omega} p(W|\mathbf{D}, \mathbf{I}) = \arg \max_{W \in \Omega} p(\mathbf{D}, \mathbf{I}|W)p(W). \quad (9)$$

In practice, two regions R_i, R_j may share the same surface model but with different reflectance models, that is, $(\ell_i^D, \Theta_i^D) = (\ell_j^D, \Theta_j^D)$ but $(\ell_i^I, \Theta_i^I) \neq (\ell_j^I, \Theta_j^I)$. For example, a picture or a piece of cloth hung on a wall or a thin book or paper on a desk, may fit to the same surfaces as the wall or desk respectively, but they have different reflectances. It is also possible that $(\ell_i^D, \Theta_i^D) \neq (\ell_j^D, \Theta_j^D)$ but $(\ell_i^I, \Theta_i^I) = (\ell_j^I, \Theta_j^I)$. To minimize the coding length and to pool information from pixels over large areas, we shall allow adjacent regions to share either depth or reflectance models. Thus, a boundary between two regions could be labeled as a *reflectance boundary*, a *depth boundary*, or both. In the following, we briefly describe the likelihood model $p(\mathbf{D}, \mathbf{I}|W)$ and the prior probability $p(W)$.

3.2 Likelihood Coupling a Mixture of Surface and Reflectance Models

In the literature, there are many ways for representing a surface, such as implicit polynomials [5], [19], superquadrics, and other deformable models. In this paper, we choose five types of generative surface models to account for various shapes in natural scenes. New models can be added under the same formulation and algorithm.

1. Family D_1 : planar surfaces with unit normal (a, b, c) and intercept d , $ax + by + cz = d$; $a^2 + b^2 + c^2 = 1$. We denote the space of all planes by $\Omega_1^D \ni \Theta$ with $\Theta = (a, b, d)$ being the plane parameters.
2. Family D_2 : conic surfaces—spheres, ellipsoids, cylinders, cones, and tori for many man-made objects and parts. We adopt the representation in [28]. These surfaces are specified by seven parameters $\Theta = (\rho, \varphi, \vartheta, k, s, \sigma, \tau)$. We refer to [28] for

detailed discussions and fitting methods. We denote by $\Omega_2^D \ni \Theta$ the space of family D_2 .

3. Family D_3 : B-spline surfaces with four control points. As surfaces in a natural scene have a broad range of sizes and orientation, we choose a reference plane $\rho : ax + by + cz = d$ which approximately fits the surface normal. Then, a rectangular domain $[0, \delta] \times [0, \phi]$ is adaptively defined on ρ to just cover the surface indexed by two parameters (u, v) , because a domain much larger than the surface will be hard to control. A grid of $h \times w$ control points are chosen on this rectangular domain. A B-spline surface is

$$s(u, v) = \sum_{s=1}^h \sum_{t=1}^w p_{s,t} B_s(u) B_t(v),$$

where $p_{s,t} = (\eta_{s,t}, \zeta_{s,t}, \xi_{s,t})$ is a control point with $(\eta_{s,t}, \zeta_{s,t})$ being coordinates on ρ and $\xi_{s,t}$ is the degree of freedom at a point. By choosing $h = w = 2$, a surface in D_3 is specified by nine parameters $\Theta = (a, b, d, \delta, \phi, \xi_{0,0}, \xi_{0,1}, \xi_{1,0}, \xi_{1,1})$. We denote by $\Omega_3^D \ni \Theta$ the space of family D_3 .

4. Family D_4 : B-spline surfaces with nine control points. Like D_3 , it consists of a reference plane ρ and a 3×3 grid. It is specified by 14 parameters $\Theta = (a, b, d, \delta, \phi, \xi_{0,0}, \dots, \xi_{2,2})$.
5. Family D_5 : cluttered surfaces. Some objects in natural scenes, such as trees and bushes, have very noisy range depth. There are no effective models in the literature for such surfaces. We adopt a nonparametric 3D histogram for the 3D points in the region. It is specified by $\Theta = (h_1^u, h_2^u, \dots, h_{L_u}^u, h_1^v, h_2^v, \dots, h_{L_v}^v, h_1^w, h_2^w, \dots, h_{L_w}^w)$, where L_u, L_v , and L_w are the number of bins in the u, v , and w directions, respectively. We denote by $\Omega_5^D \ni \Theta$ the space of family D_5 .

Fig. 5 displays some typical surfaces for the five families.

For the reflectance image \mathbf{I} , we use three families of models, denoted by $\Omega_i^I, i = 1, 2, 3$, respectively.

1. Family \mathbf{I}_1 : regions with constant reflectance $\Theta = \mu \in \Omega_1^I$. They represent most of the surfaces with uniform material properties or surfaces where range data are missing.
2. Family \mathbf{I}_2 : regions with smooth variation of reflectance, modeled by a B-spline model as in family D_3 .
3. Family \mathbf{I}_3 : This is a cluttered region with a nonparametric 1D histogram $\Theta = (h_1, h_2, \dots, h_L)$ for its intensity, with L being the number of bins.

For the surface and reflectance models above (except the histogram models), the likelihood model for a solution W assumes the fitting residues to be Gaussian noise subject to some robust statistics treatment [6], so the likelihood is

$$\begin{aligned}
 p(\mathbf{D}, \mathbf{I} | W) &= \prod_{n=1}^K p(D_{R_n}, \mathbf{I}_{R_n} ; (\ell_n^D, \Theta_n^D), (\ell_n^I, \Theta_n^I)) \quad (10) \\
 &= \prod_{n=1}^K C_n \exp \left\{ - \sum_{(i,j) \in R_n} E(D(i,j), \mathbf{I}(i,j); \right. \\
 &\quad \left. (\ell_n^D, \Theta_n^D), (\ell_n^I, \Theta_n^I)) \right\}. \quad (11)
 \end{aligned}$$

C_n is a constant depending on the model in region R_n . At each pixel (i, j) in R_n , the data energy $E_{i,j} = E(D(i, j), \mathbf{I}(i, j); (\ell_n^D, \Theta_n^D), (\ell_n^I, \Theta_n^I))$ is the squared distance from the 3D point $\mathbf{D}(i, j) = (x(i, j), y(i, j), z(i, j))$ to the fitting surface $S(\ell_n^D, \Theta_n^D)$ plus the fitness distance of reflectance $\mathbf{I}(i, j)$ to the reflectance model $J(\ell_n^I, \Theta_n^I)$.

$$\begin{aligned}
 E_{i,j} &= d^2(\mathbf{D}(i, j), S(\ell_n^D, \Theta_n^D)) \cdot \delta(\mathbf{I}(i, j) \geq \tau) \\
 &\quad + d^2(\mathbf{I}(i, j), J(\ell_n^I, \Theta_n^I)).
 \end{aligned}$$

The depth data $D(i, j)$ is considered missing if the reflectance $\mathbf{I}(i, j)$ is lower than a threshold τ , i.e., $\delta(\mathbf{I}(i, j) \geq \tau) = 0$. Such $D(i, j)$ will not be used.

In practice, we use a robust statistics method to handle outliers [6] which truncates points that are less than 25 percent of the maximum error. We further adopt the least median of squares method based on orthogonal distance [38]. Alternative likelihood models for laser radar range data [16], [17] could also be used in our framework.

3.3 Prior Models on Surfaces, Boundaries, and Corners

In general, the prior model $p(W)$ should penalize model complexity, enforce stiffness of surfaces, enhance smoothness of the boundaries, and form canonical corners at junctions. In this paper, the prior model for W is

$$p(W) = p(K)p(\pi_K) \prod_{n=1}^K p(\ell_n^D)p(\Theta_n^D | \ell_n^D)p(\ell_n^I)p(\Theta_n^I | \ell_n^I).$$

$\pi_K = (R_1, \dots, R_K)$ denotes a K -partition of the lattice Λ which forms a planar graph with K faces for the regions, M edges for boundaries, and N vertices for corners,

$$\pi_K = (R_k, k = 1, \dots, K; \Gamma_m, m = 1, \dots, M; V_n, n = 1, \dots, N).$$

We treat energy terms on R_k , Γ_m , and V_n independently, thus

$$p(\pi_K) = \prod_{k=1}^K p(R_k) \prod_{m=1}^M p(\Gamma_m) \prod_{n=1}^N p(V_n).$$

Our prior probability $p(W)$ consists of four parts.

1. *Prior on surface number and area sizes.* It is well-known that a higher order model always fits a surface better than a lower order model, but the former could be less stable in the presence of noise. Some conventional model complexity criteria in model selection and merging techniques include MDL (minimum description length) [34], AIC (Akaike Information Criterion), BIC (Bayesian information criterion) [1]. A survey study for range surface fitting is reported in [7]. According to such criteria, model complexity is regularized by three factors, which penalizes the number of surfaces K and the number of parameters in each surface model $\#\Theta$, respectively.

$$\begin{aligned}
 p(K) &\propto e^{-\lambda_0 K}, \quad p(\Theta_n^D | \ell_n^D) \propto e^{-\lambda^D \#\Theta_n^D}, \\
 \text{and } p(\Theta_n^I | \ell_n^I) &\propto e^{-\lambda^I \#\Theta_n^I}, \quad \forall n. \quad (12)
 \end{aligned}$$

However, in our experiments as well as in our previous work [36], we observed that such criteria are not appropriate in comparison with human segmentation results. Conventional model complexity criteria, like MDL, are motivated by shortest image coding. But, the task of segmentation and image understanding is very different from coding. The extent to which an object is segmented depends on the importance and familiarity of the object in the scene and the task. In particular, natural scenes contain a very broad range of sizes measured by their areas. Unfortunately, it is impractical to define the importance of each type of object in a general purpose segmentation algorithm, so we adopt a statistical model on the surface areas $|R_n|$

$$p(R_n) \propto e^{-\alpha |R_n|^c}, \quad \forall n = 1, 2, \dots, K, \quad (13)$$

where c is a constant and α is a scale factor to control the scale of the segmentation. In our experiments, α is the only parameter that is left to be adjusted. All other parameters are set to a single value for all experiments.

2. *Prior on B-spline control points for surface stiffness.* For all the B-spline models, a prior is imposed on the control points $\{\xi_{s,t} : 0 \leq s, t \leq 2 \text{ or } 3\}$ such that the surface is as planar as possible. We triangulate the spline grid on the ρ -plane and every adjacent three control points form a plane. The prior energy terms enforce the normals of adjacent planes to be parallel to each other. A similar prior was used in the wireframe reconstruction in [25].
3. *Prior for surface boundary smoothness.* Due to the heavy noise of the range data along surface boundaries, a boundary smoothness prior is adopted, like in the SNAKE [23] or region competition model [39]. Let $\Gamma(s) = (x(s), y(s))$, $s \in [a, b]$ be a boundary between two surfaces,

$$\begin{aligned}
 p(\Gamma(s)) &\propto \exp \left\{ - \int \phi(\dot{\Gamma}(s)) + \phi(\ddot{\Gamma}(s)) ds \right\}, \quad \text{or} \\
 p(\Gamma(s)) &\propto \exp \left\{ - \int \sqrt{\dot{x}^2(s) + \dot{y}^2(s)} ds \right\}.
 \end{aligned}$$

$\phi()$ is a quadratic function with flat tails to account for sharp L -shaped turns in boundaries. We choose the second one as the smoothness prior.

4. *Prior for canonical corners.* On each vertex V_n , we impose a prior model $p(V_n)$, $n = 1, 2, \dots, N$ adopted from the line-drawing work in [25]. It assumes that the angles at a corner should be more or less equal.

To summarize, the Bayesian framework provides a convenient way for integrating multiple generative models, for coupling two cues, and for introducing prior models. This enables us to deal with complex natural scenes.

4 EXPLORING THE SOLUTION SPACE BY JUMP-DIFFUSION

As we illustrated in the 1D case, the posterior probability is distributed over a countable number of subspaces of varying dimensions. In the range segmentation literature, some methods, such as edge detection [24], region growing [19], [5], clustering [20], [12], and generalized Hough transforms can produce useful information, but none of these methods are capable of exploring such complex spaces thoroughly.

Our algorithm extends the 1D range examples in Section 2. It consists of six types of Markov chain jumps and a boundary diffusion processes. To speed up the MCMC search, we use data clustering in each model space and an edge detection/partition on the lattice Λ . These are discussed in the following three sections.

4.1 Six Dynamics with Jumps and Diffusion

This section briefly presents the six types of moves/dynamics which form a Markov chain that can traverse the solution space.

Dynamics 1: diffusion of region boundary—stochastic region competition. Within a subspace of fixed dimension (i.e., the number of surfaces and their models are given), the boundaries evolve according to a stochastic version of the region competition equations [39]. Let $\Gamma_{ij}(s) = (x(s), y(s))$, $s \in (a, b)$ denote the boundary between two regions R_i and R_j , and let $(\ell_i^D, \Theta_i^D, \ell_i^I, \Theta_i^I)$ and $(\ell_j^D, \Theta_j^D, \ell_j^I, \Theta_j^I)$ be the models of the two regions, respectively. The motion of curve $\Gamma_{ij}(s)$ follows the following equation [39].

$$\begin{aligned} \frac{d\Gamma_{ij}(s)}{dt} = & \\ & - \frac{\delta \log p(W|\mathbf{D}, \mathbf{I})}{\delta \Gamma_{ij}(s)} + \sqrt{2T(t)} dw_t \bar{\mathbf{n}}(s), \quad dw_t \sim N(0, (dt)^2). \end{aligned} \quad (14)$$

The Brownian motion is always along the curve normal direction $\bar{\mathbf{n}}(s) = \frac{(-\dot{y}(s), \dot{x}(s))}{\sqrt{\dot{x}^2(s) + \dot{y}^2(s)}}$.

To couple with the continuous representation of curves Γ_{ij} , we assume the lattice Λ to be a continuous 2D plane. The curve $\Gamma_{ij}(s)$ is involved in three terms in the posterior $p(W|\mathbf{D}, \mathbf{I})$: the smoothness prior and the likelihood on two regions R_i and R_j .

By a Green's theorem and an Euler-Lagrange equation, the gradient is

$$\begin{aligned} \frac{d\Gamma_{ij}(s)}{dt} = & \left\{ -2\mu\kappa(s) + \log \frac{p(\mathbf{D}(x(s), y(s)); (\ell_i^D, \Theta_i^D))}{p(\mathbf{D}(x(s), y(s)); (\ell_j^D, \Theta_j^D))} \right. \\ & \cdot \delta(\mathbf{I}(x(s), y(s)) \geq \tau) + \log \frac{p(\mathbf{I}(x(s), y(s)); \ell_i^I, \Theta_i^I)}{p(\mathbf{I}(x(s), y(s)); \ell_j^I, \Theta_j^I)} \\ & \left. + \sqrt{2T(t)} \frac{dw_t}{dt} \right\} \bar{\mathbf{n}}(s). \end{aligned}$$

In the above equations, $\kappa(s)$ is its curvature. At each point $(x(s), y(s))$ along the curve, two local loglikelihood ratio tests compare the fitness of the two region models: one for the surface model and the other for the reflectance model. When the range data is less reliable, i.e., $\delta(\mathbf{I}(x(s), y(s)) \geq \tau) = 0$, its loglikelihood ratio test is not used. For clarity, we omit another energy term originated from the prior in (13) on the areas $|R_i|$ and $|R_j|$.

Dynamics 2: diffusion of vertices. A vertex $V = (x, y)$ refers to an intersection of more than two regions. It involves some prior model $p(V)$ for canonical corners in the previous section, and the curvature is ill-defined at such a point. Its diffusion is implemented by the Gibbs sampler [13]. That is, we consider a local lattice, say 3×3 pixels, and randomly select a position subject to the posterior probability.

Dynamics 3: diffusion of surface and reflectance models. The diffusion of the parameters $((\Theta_n^D, \Theta_n^I))$ for a region R_n , $n = 1, 2, \dots, K$, with other variables in W fixed, is:

$$\frac{d(\Theta_n^D, \Theta_n^I)}{dt} = \frac{d \log p(\mathbf{D}_{R_n}, \mathbf{I}_{R_n}; \ell_n^D, \Theta_n^D, \ell_n^I, \Theta_n^I)}{d(\Theta_n^D, \Theta_n^I)}.$$

Robust statistics methods are used in calculating the gradient. Some range pixels do not contribute to the surface fitting if the reflectance is low. In experiments, we found that the Brownian motion does not amount to any noticeable improvement in the above equation in the parameter space; hence, we leave it out in practice.

Dynamics 4: switching a surface or reflectance model ℓ_n^D or ℓ_n^I . This is similar to the 1D example, but we have more families of model to choose from. Suppose, at a time instance, a region R_n is selected to switch to a model ℓ_n^D . Then, we need some heuristic information for the new model Θ_n^D . The importance proposal probability is calculated, like $q(\phi|m)$ in (3), based on a number of candidate surfaces pre-computed by a data clustering approach. As we shall discuss below, data clustering is a better method than the Hough transform in high-dimensional spaces.

Dynamics 5 and 6: splitting and merging of regions. Split and merge are a pair of reversible moves to realize the jump process between subspaces. Suppose that a region R_k with model (Θ_k^D, Θ_k^I) is split into two regions R_i and R_j with models (Θ_i^D, Θ_i^I) and (Θ_j^D, Θ_j^I) , respectively; the present state W and the new state W' are

$$\begin{aligned} W = & (K, R_k, (\ell_k^D, \Theta_k^D), (\ell_k^I, \Theta_k^I)), W_-, \\ W' = & (K + 1, R_i, R_j, (\ell_i^D, \Theta_i^D), (\ell_i^I, \Theta_i^I), (\ell_j^D, \Theta_j^D), (\ell_j^I, \Theta_j^I), W_-). \end{aligned}$$

W_- includes all other variables in W that remain unchanged during this jump. The split and merge are proposed with probability $G(W \rightarrow W')dW'$ and $G(W' \rightarrow W)dW$, while the split move is accepted with probability

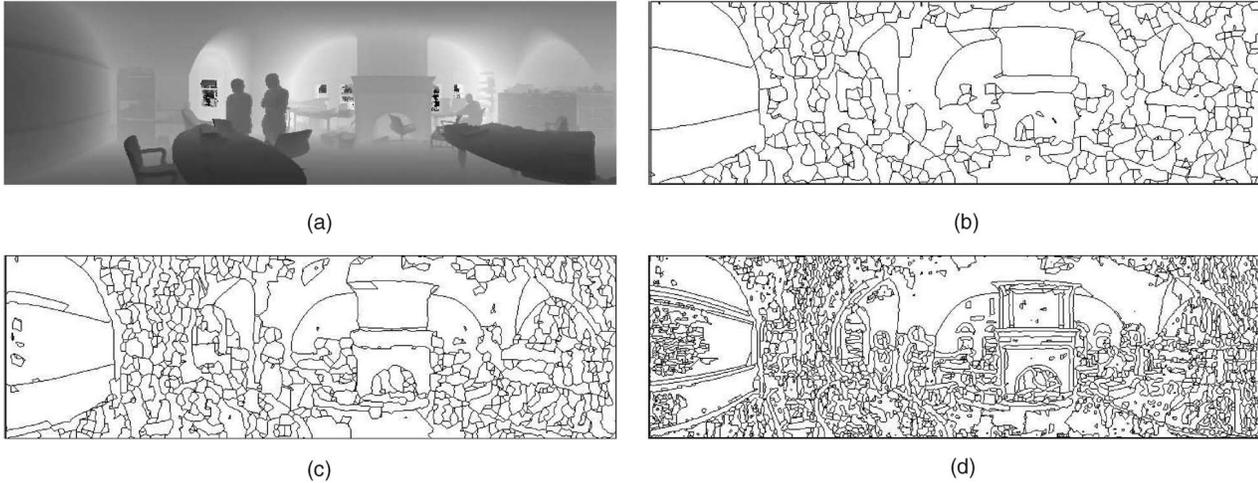


Fig. 6. Computed edge maps at three scales for an office scene. (a) Range image of an office scene. (b) Edge detection and partition scale 1. (c) Edge detection and partition scale 2. (d) Edge detection and partition scale 3.

$$\alpha(W \rightarrow dW') = \min\left(1, \frac{G(W' \rightarrow dW)p(W'|\mathbf{I})dW'}{G(W \rightarrow dW')W'p(W|\mathbf{I})dW}\right). \quad (15)$$

The merge proposal probability is,

$$G(W' \rightarrow dW) = q(6)q(R_i, R_j)q(\ell_k^D, \Theta_k^D | R_k)d\Theta_k^D q(\ell_k^I, \Theta_k^I | R_k)d\Theta_k^I. \quad (16)$$

$q(6)$ is the probability for choosing the merge move, and $q(R_i, R_j)$ is the probability for choosing R_i, R_j . $q(\ell_k^D, \Theta_k^D | R_k)$ is the probability for a new surface model. We first compute an excessive number of candidates by a data clustering method, like the Hough transform for the 1D case. Then, each candidate model is voted by the pixels inside the region R_k and, thus, a nonparametric proposal probability is constructed for $q(\ell_k^D, \Theta_k^D | R_k)$ with weights proportional to the number of votes.

Similarly, the split proposal probability is $G(W \rightarrow dW') =$

$$q(5)q(R_k)q(\Gamma_{ij}|R_k)q(\ell_i^D, \Theta_i^D | R_i)d\Theta_i^D q(\ell_i^I, \Theta_i^I | R_i)d\Theta_i^I q(\ell_j^D, \Theta_j^D | R_j)d\Theta_j^D q(\ell_j^I, \Theta_j^I | R_j)d\Theta_j^I.$$

Once R_k is chosen to split, Γ_{ij} is a candidate splitting boundary. In the 1D example, this is randomly chosen by an edge strength function. In 2D, this is selected from a set of candidate partitions precomputed by edge detection.

In the following, we focus on the computation of two importance proposal probabilities used above: 1) $q(\Gamma | R)$ —splitting the boundary of a region R . 2) $q(\ell, \Theta | R)$ —the new model of a region (surface or reflectance).

4.2 Coarse-to-Fine Edge Detection and Partition

In this section, we detect potential surface boundaries based on local edge cues, and trace the edges to form a partition of the lattice which will be used as candidate boundaries in the split-merge jumps. Since natural scenes contain objects in a broad range of sizes, we organize the edge maps in three scales according to an edge strength measure that we shall discuss below. For example, Fig. 6 displays one example for

an office scene in which the edges are based on both range and reflectance measures.

To overcome noise, we first estimate the surface normal at each point $(x, y) \in \Lambda$ over a small window Δ , say 5×5 pixels. Let $\{p_i = (x_i, y_i, z_i) : (m, n) \in \Delta(x, y), i = 1, 2, \dots, |\Delta|\}$ be a set of 3D points in a local patch Δ centered at (x, y) , and \bar{p} be their mass center. One can estimate the local surface normal by minimizing the quadratic error function [12]

$$\mathbf{n}^* = \arg \min_{\mathbf{n}} \mathbf{n}' S \mathbf{n}, \quad \text{with } S = \sum_i (p_i - \bar{p})(p_i - \bar{p})',$$

where \mathbf{n}^* is the eigenvector of the scatter matrix S that corresponds to the smallest eigenvalue.

With the normal \mathbf{n} , we fit a local plane $ax + by + cz = d$ ($c = \sqrt{1 - a^2 - b^2}$) to the patch, and obtain a vector representation $= (a, b, d)$ for each point (x, y) . An edge strength is computed on this vector field using a technique from [30]. We compute a 2×2 matrix at each point (x, y) ,

$$\Sigma(x, y) = \int \int_{\Delta(x, y)} \begin{pmatrix} \nabla_x^2 & \nabla_x \nabla_y \\ \nabla_x \nabla_y & \nabla_y^2 \end{pmatrix} \rho(u - x, v - y) dudv, \quad (17)$$

where $\rho(u - x, v - y)$ is a Parzen window centered at (x, y) . Let λ_1 and λ_2 ($\lambda_2 \leq \lambda_1$) be the two eigenvalues of the matrix, and v_1 and v_2 be the corresponding eigenvectors. Then, the edge strength, orientation, and cornerness are measured by e, θ , and c , respectively,

$$e(x, y) = \sqrt{\lambda_1 + \lambda_2}, \quad \theta(x, y) = \arg(v_1), \quad c(x, y) = \lambda_2.$$

In addition to computing the edge maps from range images, we also apply standard edge detection to the reflectance image and obtain edge maps on three scales. We threshold the edge strength $e(x, y)$ at three levels to generate the edge maps shown in Fig. 6 after tracing them with heuristic local information to form closed partitions [10].

Choosing a region R to split, we superimpose R with one of the three edge maps depending on the size of R (large regions will use coarse edge partitions in general). Then, the edge partitions within R are candidate subregions. Thus, the splitting boundaries Γ are chosen at random from a set

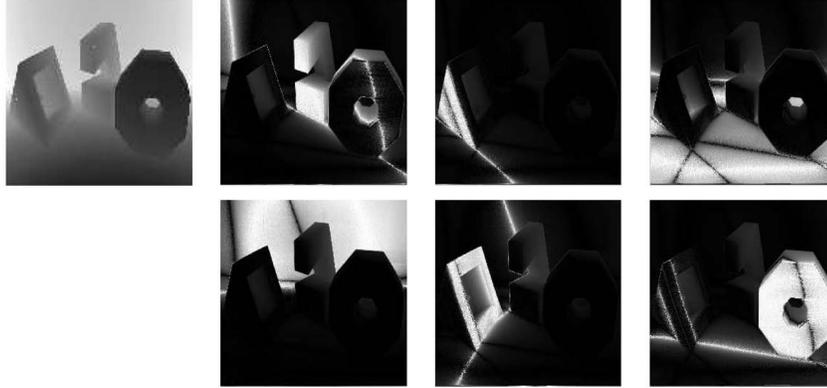


Fig. 7. A polyhedra range image with six saliency maps for six clustered surfaces.

of candidates. It is worth mentioning that these partitions are only heuristics for proposal probabilities. Some mistakes on the boundary can still be corrected by the boundary diffusion steps. As an alternative, we can always use the original lattice as the partition at scale 4, which only slows down the algorithm. We refer to our previous work [36] for a detailed formulation, and a more rigorous treatment is recently developed in [4] which transfers the split-merge jumps into a graph partition problem and designs the split-merge by a Swendsen-Wang cut method.

4.3 Coarse-to-Fine Surface Clustering

We compute importance proposal probabilities on the parameter spaces Ω_1^D , Ω_2^D , Ω_3^D , Ω_4^D , and Ω_5^D , respectively. These probabilities are expressed by a set of candidate surface models in nonparametric forms. But, unlike the 1D example presented in Section 2, we shall use data clustering instead of Hough transforms for two reasons: 1) Hough transforms become impractical in high-dimensional space (say more than three dimensions) and 2) Hough transforms assume a 2-category detection and, thus, the peaks (candidates) in the space can be contaminated by each other. In contrast, data clustering is more general.

In the edge detection step, we have fitted each small patch Δ to a local plane (a, b, d) with mass center \bar{p} and the smallest eigenvalue λ_{\min} of the scatter matrix S . Therefore, we collect a set of patches by subsampling the lattice Λ by a factor of δ for computational efficiency,

$$Q = \{(\Delta_j, a_j, b_j, d_j, \bar{p}_j, \lambda_{\min,j} : j = 1, 2, \dots, J = |\Lambda|/\delta^2)\}.$$

We can discard patches which have relatively large λ_{\min} , i.e., patches that are likely on the boundary.

The patches in set Q are clustered into a set of C candidate surfaces in all five model spaces

$$\mathcal{C} = \{\Theta_i : \Theta_i \in \Omega_1^D \cup \Omega_2^D \cup \Omega_3^D \cup \Omega_4^D \cup \Omega_5^D, i = 1, \dots, C\},$$

where C is chosen to be excessive. We use the EM-clustering algorithm to compute the candidate surfaces. Alternatively, one may use the mean-shift clustering method [8], [9].

For example, Fig. 7 shows six selected clusters (among many) for a polyhedra scene. Each cluster is associated with a ‘‘saliency map’’ where the brightness at a patch displays the

probability that it fits to the cluster (or candidate model). Such probability comes automatically from the EM-clustering. It is very informative in such simple scenes where the models are sufficient to describe the surfaces and objects have similar sizes.

In natural scenes, the results are less impressive. Very often small objects, like books on the top of a desk, are relatively too small to form a separate cluster in the presence of other large objects (clusters) like wall and ceiling. To resolve this problem, we compute the clusters in a coarse-to-fine strategy. For example, Fig. 8 shows eight chosen saliency maps for the most prominent clusters in the office scene, which correspond to the floor, desktop, furnace, windows, walls, and ceiling, respectively. The total sum of the probability over the lattice is a measure of how prominent a cluster is. Then, for patches in Q that do not fit very well to these prominent clusters, we refine the range data by subpixel interpolation, and conduct the clustering on such areas. For example, the lower panel in Fig. 8 displays six of the clusters for a subarea (on the left, indicated by windows in the big scene above), such as, people, chairbacks, small box, etc.

These candidate models are used to form the importance proposal probabilities as in the 1D example. Given a region R , each pixel inside R votes for the candidate models by a continuous probability. Then, the proposed model is selected from the candidates proportional to their votes and some random perturbations.

5 EXPERIMENTS

5.1 The Data Sets and Preprocessing

We test the algorithm on three range data sets. The first two are the standard Perceptron LADAR camera images and K2T structured light camera images in the USF data set. The third one is a data set from Brown University, where images are collected with a long range scanner LMS-Z210 by Riegl. The field of view is 80 degrees vertically and 259 degrees horizontally. Each image contains 444×1440 measurements with an angular separation of 0.18 degree.

As range data are contaminated by heavy noise, some preprocessing algorithm is helpful to deal with various errors present in the data, while preserving the discontinuities. In our experiments, we adopt the least median of



Fig. 8. Saliency maps for an office scene at two scales. See text for explanation. (a) Range image. (b) Reflectance image.

squares (LMedS) and anisotropic diffusion [37] sequentially to remove noise. LMedS is related to the median filter used in image processing to remove impulsive noise from images and can be used to remove strong outliers in range data. The anisotropic diffusion removes small noise while preserving the discontinuities along the surface boundaries.

5.2 Results and Evaluation

We fix the parameters in the algorithm to be the same for all three data sets with only one free parameter c in (13). This parameter controls the extent of the segmentation. Even in human segmentation of the same scene, people may come up with coarse or very detailed segmentations. The algorithm starts with arbitrary initializations: a 2×2 grid partition or randomized initializations.

Fig. 9 displays the segmentation results on four images in data set 1. Fig. 10 shows two examples from data set 2. For these two data sets, we only use range data and the segmentations are superimposed on the reflectance images. For comparison, we also show a manual segmentation used in [21] and [32]. We also show the 3D reconstructed scenes from a novel viewing angle using OpenGL based on our segmentation results and the fitted surface models. This is a good way to examine the sufficiency of the models used. In these reconstructed 3D scenes, the background and floor behind the occluding objects are completed using the simple method discussed below. It is not surprising that the algorithm can parse such scenes very well because the image models are sufficient to account for the surfaces in these two data sets.

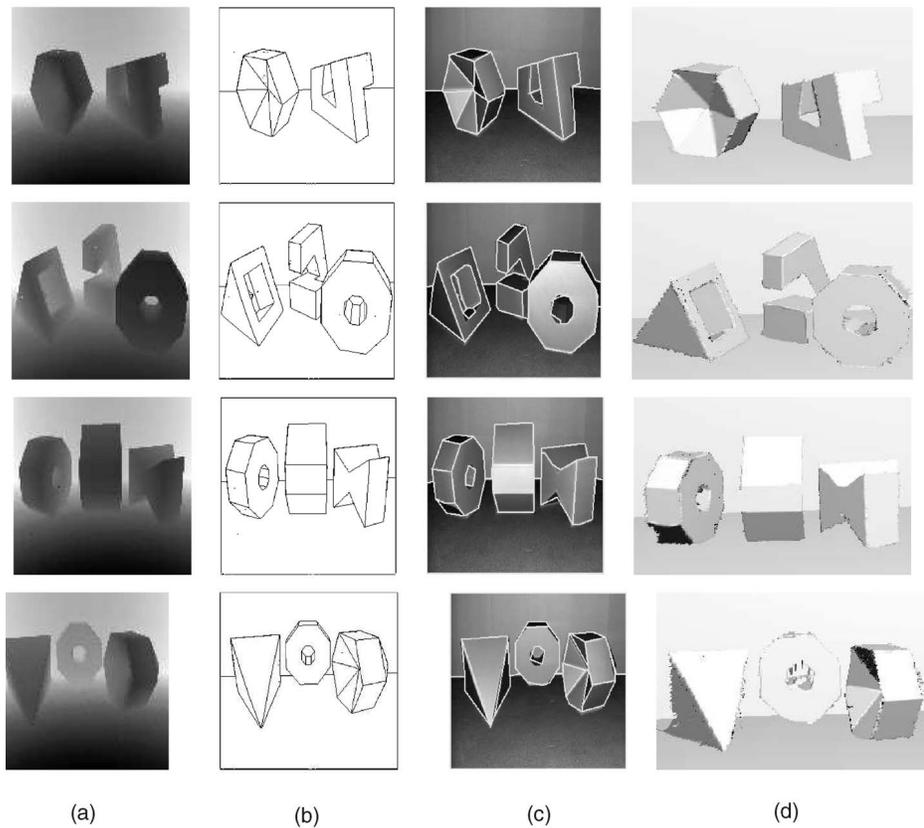


Fig. 9. Segmentation results compared with the manual segments provided in [21]. We only use range data and the segments are superimposed on the reflectance images in (c). The reconstructions are shown in slightly different views. (a) Range data. (b) Manual segment. (c) Our result. (d) 3D reconstruction.

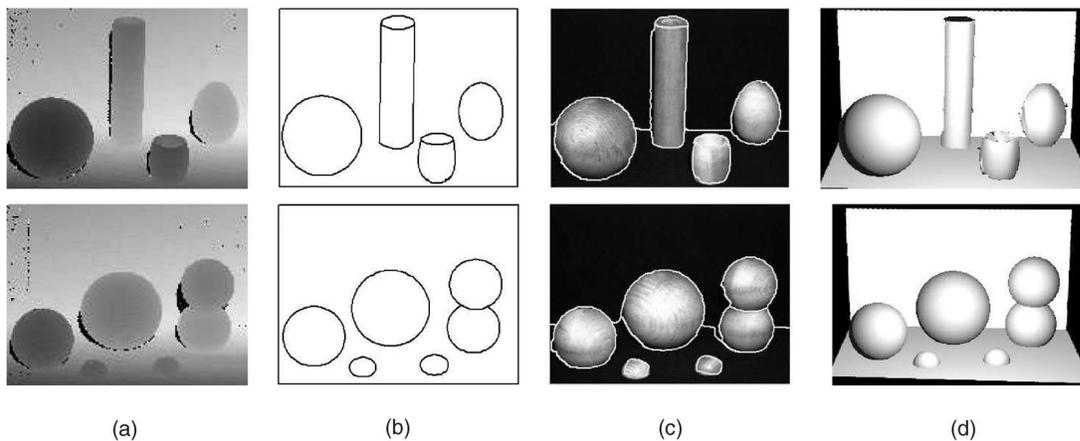


Fig. 10. Segmentation on the second data set compared with manual segmentations provided in [32]. We only use range data and the segmentations are superimposed on the reflectance images in (c). The reconstruction is generated from novel views. (a) Range. (b) Manual segment. (c) Our result. (d) 3D reconstruction.

Six examples from the Brown data set are shown in Figs. 11 and 12. For lack of space, we only show small part of each scene. The trees in these scenes are correctly segmented and thus it seems that the clutter model does well for such cluttered regions.

Object surfaces in range images are often incomplete due to partial occlusion and poor surface reflectance at various places. In the 3D scene reconstruction stage, it is important to fill the missing data points and complete surfaces behind occlusions [11]. The completion of depth information needs higher-level understanding of the 3D objects. In our

experiments, we use a simple method which proceeds in two steps: 1) Identify the types of boundaries, such as creasing and occluding, based on the surface functions and 2) compute the ownership of the boundary, i.e., which surface is in the front. Once the surface order is computed, we complete the occluded areas by extending the surface function fitted to that region. This works well for the walls and the floor as we showed in the 3D reconstructed scenes in Figs. 9 and 10. But, it is much more challenging to complete surfaces in the Brown data set.

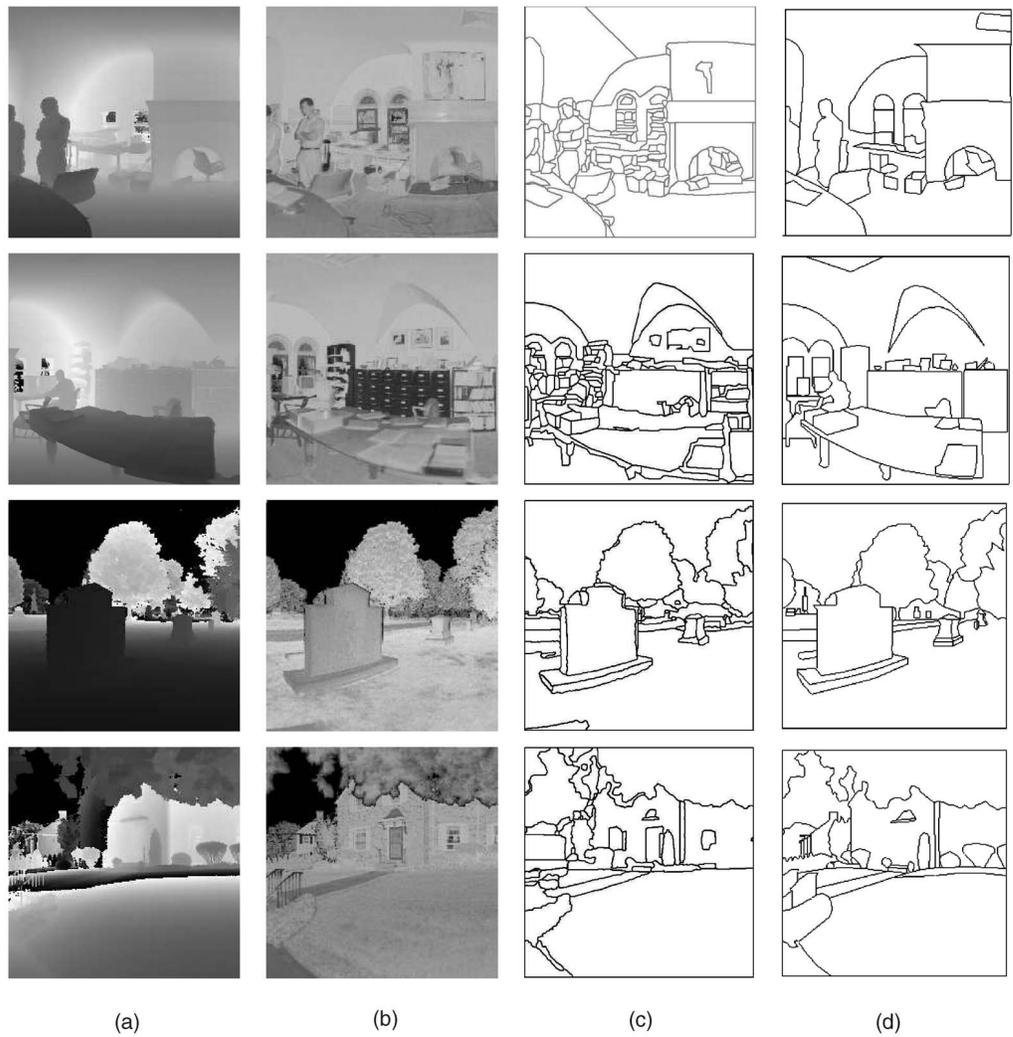


Fig. 11. Segmentation results for parts of the four scenes in Fig. 1. (a) Range. (b) Reflectance. (c) Our result. (d) Manual segment.

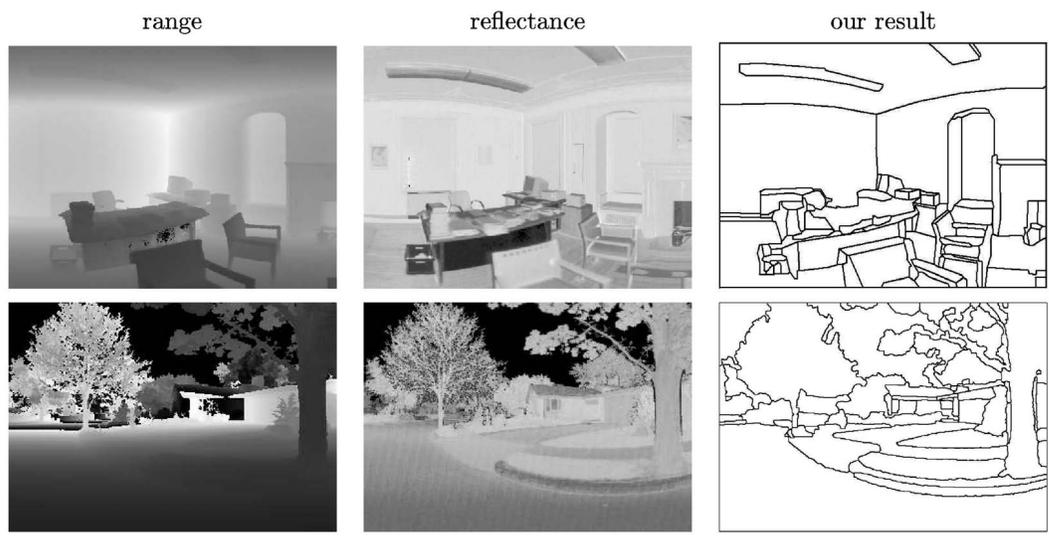


Fig. 12. Segmenting the most cluttered part of office B in Fig. 1 and a scene with trees.

6 DISCUSSION

We adopt a Bayesian formulation to integrate two cues, many families of generative models, and a number of prior models. We use a jump-diffusion process as a general tool for designing Markov chains traversing the complex solution space so that the solution no longer depends on initialization. The computational speed is largely expedited by bottom-up heuristics.

Some remaining problems that need to be resolved in future research include:

1. The algorithm is still time consuming and takes about 1 hour on a Pentium IV PC to segment a scene (usually having 300×300 pixels) with arbitrary initial conditions. Recent developments using the Swendsen-Wang cut [4] suggest considerable improvements in speed.
2. The experiments reveal that when the models are not sufficient, the segmentation is not good. For example, the cable in the air in Fig. 11 is missing because of being a 1D structure, not a region.
3. Better prior models for 3D objects are needed to group surfaces into objects and, therefore, to complete surfaces behind the occluding objects.
4. We acknowledge that the manual segmentations used for evaluation may not correspond to the optimal solutions given by the limited models and it remains an open question for measuring the goodness of a segmentation solution.

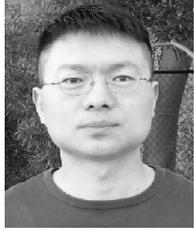
ACKNOWLEDGMENTS

The authors acknowledge the use of the range data sets from the vision group at University of Southern Florida. They would like to thank Ann Lee and David Mumford for their generous assistance with the Brown data set. They also appreciate an MS student Qiming Luo for implementing the 1D examples which first appeared as a master's thesis [26] supervised by S.C. Zhu. This work is supported by US Office of Naval Research grant N00014-02-1-0952 and US National Science Foundation grant IIS-0244763.

REFERENCES

- [1] H. Akaike, "A Bayesian Analysis of the Minimum AIC Procedure," *Annals of the Inst. Statistical Math.*, vol. 30A, pp. 9-14, 1978.
- [2] F. Arman, B. Sabata, and J.K. Aggarwal, "Segmentation of 3D Range Images Using Pyramidal Data Structures," *Proc. CVGIP: Image Understanding*, vol. 57, no. 3, pp. 373-387, 1993.
- [3] D.H. Ballard, "Generalized Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111-122, 1981.
- [4] A. Barbu and S.C. Zhu, "Graph Partition by Swendsen-Wang Cuts," *Proc. Int'l Conf. Computer Vision*, 2003.
- [5] P.J. Besl and R.C. Jain, "Segmentation through Variable Order Surface Fitting," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 2, pp. 167-192, 1988.
- [6] M.J. Black and A. Rangarajan, "On the Unification of Line Process, Outlier Rejection, and Robust Statistics with Applications in Early Vision," *Int'l J. Computer Vision*, vol. 19, no. 1, pp. 57-91, Jan. 1996.
- [7] K. Bubna, "Model Selection Techniques and Merging Rules for Range Data Segmentation Algorithms," *Computer Vision and Image Understanding*, vol. 80, pp. 215-245, 2000.
- [8] Y. Cheng, "Mean Shift, Mode Seeking, and Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790-799, Aug. 1995.
- [9] D. Comaniciu and P. Meer, "Mean Shift Analysis and Applications," *Proc. Int'l Conf. Computer Vision*, pp. 1197-1203, 1999.
- [10] I.J. Cox, J.M. Rehg, and S. Hingorani, "A Bayesian Multiple-Hypothesis Approach to Edge Grouping and Contour Segmentation," *Int'l J. Computer Vision*, vol. 11, no. 1, pp. 5-24, 1993.
- [11] F. Dell'Acqua and R. Fisher, "Reconstruction of Planar Surfaces Behind Occlusions in Range Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 569-575, Apr. 2002.
- [12] P.J. Flynn and A.K. Jain, "Surface Classification: Hypothesis Testing and Parameter Estimation," *Proc. Computer Vision and Pattern Recognition*, 1988.
- [13] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721-741, 1984.
- [14] S. Geman and C.R. Huang, "Diffusion for Global Optimization," *SIAM J. Control and Optimization*, vol. 24, no. 5, pp. 1031-1043, 1986.
- [15] P.J. Green, "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination," *Biometrika*, vol. 82, pp. 711-732, 1995.
- [16] T.J. Green and J.H. Shapiro, "Maximum-Likelihood Laser Radar Range Profiling with the Expectation-Maximization Algorithm," *Optical Eng.*, vol. 31, pp. 2343-2354, 1992.
- [17] T.J. Green and J.H. Shapiro, "Detecting Objects in 3D Laser Radar Range Images," *Optical Eng.*, vol. 33, pp. 865-873, 1994.
- [18] U. Grenander and M.I. Miller, "Representations of Knowledge in Complex Systems," *J. Royal Statistical Soc. Series B*, vol. 56, no. 4, 1994.
- [19] A. Gupta, A. Leonardis, and R. Bajcsy, "Segmentation of Range Images as the Search for Geometric Parametric Models," *Int'l J. Computer Vision*, vol. 14, no. 3, pp. 253-277, 1995.
- [20] R.L. Hoffman and A.K. Jain, "Segmentation and Classification of Range Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 608-620, 1987.
- [21] A. Hoover, G. Jean-Baptiste, X. Jiang, P.J. Flynn, H. Bunke, D.B. Goldgof, K. Bowyer, D.W. Eggert, A. Fitzgibbon, and R.B. Fisher, "An Experimental Comparison of Range Image Segmentation Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 673-689, July 1996.
- [22] J.G. Huang, A.B. Lee, and D.B. Mumford, "Statistics of Range Images," *Proc. Computer Vision and Pattern Recognition*, 2000.
- [23] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J. Computer Vision*, vol. 1, no. 4, pp. 321-332, 1988.
- [24] R. Krishnapuram and S. Gupta, "Morphological Methods for Detection and Classification of Edges in Range Images," *Math. Imaging and Vision*, vol. 2, pp. 351-375, 1992.
- [25] Y.G. Leclerc and M.A. Fischler, "An Optimization-Based Approach to the Interpretation of Single Line Drawings as 3D Wire Frame," *Int'l J. Computer Vision*, vol. 9, no. 2, pp. 113-136, 1992.
- [26] Q.M. Luo, "Signal and Range Image Segmentation by Data Driven Markov Chain Monte Carlo," master's thesis, Dept. of Computer and Information Science, The Ohio State Univ., 2001.
- [27] R. Maciucă and S.C. Zhu, "How do Heuristics Expedite Markov Chain Search?" *Proc. Int'l Workshop Statistical and Computational Theories of Vision*, 2003.
- [28] D. Marshal, G. Lukacs, and R. Martin, "Robust Segmentation of Primitives from Range Data in the Presence of Geometric Degeneracy," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 304-314, Mar. 2001.
- [29] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, "Equations of State Calculations by Fast Computing Machines," *J. Chemical Physics*, vol. 21, pp. 1087-1092, 1953.
- [30] M. Nitzberg, T. Shiota, and D. Mumford, "Filtering, Segmentation and Depth," *Lecture Notes in Computer Science*, vol. 662, 1993.
- [31] D.B. Phillips and A.F.M. Smith, "Bayesian Model Comparison via Jump Diffusion," *Markov Chain Monte Carlo in Practice*, W.R. Gilks, S.T. Richardson and D.J. Spiegelhalter, eds., chapter 13, Chapman and Hall, 1995.
- [32] M.W. Powell, K.W. Bowyer, X. Jiang, and H. Bunke, "Comparing Curved-Surface Range Image Segmentation," *Proc. Int'l Conf. Computer Vision*, pp. 286-291, 1998.
- [33] I. Stamos and P.K. Allen, "Geometry and Texture Recovery of Scenes of Large Scale," *Computer Vision and Image Understanding*, vol. 88, no. 2, pp. 94-118, Nov. 2002.
- [34] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*. Singapore: World Scientific, 1989.
- [35] L. Tierney, "Markov Chains for Exploring Posterior Distributions," *Annals of Statistics*, vol. 22, pp. 1701-28, 1994.

- [36] Z. Tu and S.C. Zhu, "Image Segmentation by Data-Driven Markov Chain Monte Carlo," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 657-673, May 2002.
- [37] M. Umasuthan and A.M. Wallace, "Outlier Removal and Discontinuity Preserving Smoothing of Range Data," *IEE Proc. Vision and Image Signal Process*, vol. 143, no. 3, 1996.
- [38] Z.Y. Zhang, "Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting," technical report, INRIA, Sofia, France, 1995.
- [39] S.C. Zhu and A.L. Yuille, "Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884-900, Sept. 1996.



Feng Han received the BS degree in computer science from Dalian University of Technology in 1996. He received the ME degree in electrical and electronic engineering from Nanyang Technological University in 2000. From 2000 to 2002, he studied in the Department of Computer and Information Science at The Ohio State University. He is currently a PhD candidate in the Computer Science Department at the University of California, Los Angeles (UCLA). His research

interests include computer vision, image processing, pattern recognition, and computer graphics.



Zhuowen Tu received the BE degree in electronic engineering from Beijing Information Technology Institute in 1993. He received the ME degree in civil engineering from Tsinghua University in 1996. He was with the Institute of Computer Science and Technology at Beijing University from 1996 to 1997. He received the PhD degree in computer science in 2002 from The Ohio State University. He is currently a postdoctoral fellow in the Department of Statistics at the University of California, Los Angeles. He received the Talbert Abrams award (Honorable mention) by American Society of Photogrammetry and Remote Sensing in 2003. He was also awarded with the David Marr prize at the ninth international conference on computer vision. His research interests include computer vision, image processing, pattern recognition, and digital mapping.



Song-Chun Zhu received the BS degree from the University of Science and Technology of China in 1991, and the MS and PhD degrees from Harvard University in 1994 and 1996, respectively. All degrees are in computer science. He is currently an associate professor jointly with the Departments of Statistics and Computer Science at the University of California, Los Angeles (UCLA). He is codirector of the UCLA Center for Image and Vision Science. Before joining UCLA, he worked at Brown University (applied math), Stanford University (computer science), and The Ohio State University (computer science). His research is focused on computer vision and learning, statistical modeling, and stochastic computing. He has published more than 70 papers and received a number of honors, including the David Marr prize honorary nomination in 1999, and the David Marr prize in 2003, the Sloan fellow in computer science, the US National Science Foundation Career Award, and an US Office of Naval Research Young Investigator Award.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.