A Stochastic Algorithm for 3D Scene Segmentation and Reconstruction

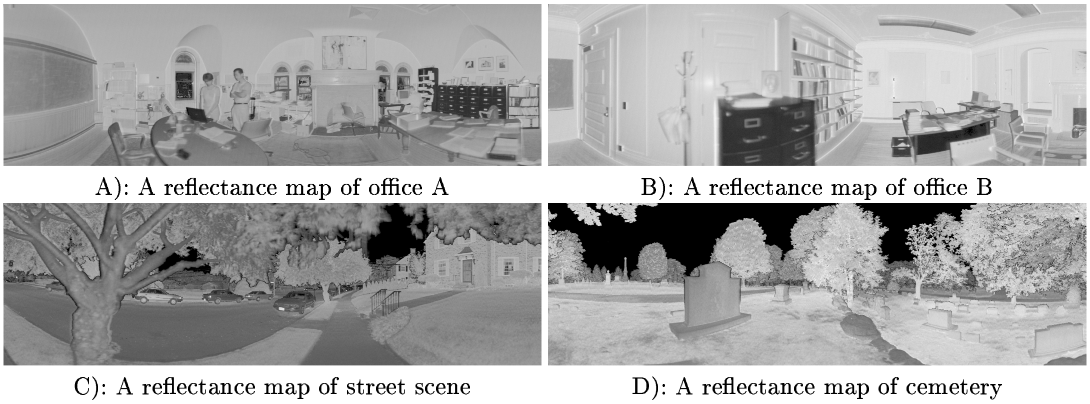
Feng Han, Zhouwen Tu, and Song-Chun Zhu

Dept. of Comp. and Info. Sci., Ohio State Univ., Columbus, OH 43210, USA
{hanf, ztu, szhu}@cis.ohio-state.edu,

Abstract. In this paper, we present a stochastic algorithm by effective Markov chain Monte Carlo (MCMC) for segmenting and reconstructing 3D scenes. The objective is to segment a range image and its associated reflectance map into a number of surfaces which fit to various 3D surface models and have homogeneous reflectance (material) properties. In comparison to previous work on range image segmentation, the paper makes the following contributions. Firstly, it is aimed at generic natural scenes, indoor and outdoor, which are often much complexer than most of the existing experiments in the "polyhedra world". Natural scenes require the algorithm to automatically deal with multiple types (families) of surface models which compete to explain the data. Secondly, it integrates the range image with the reflectance map. The latter provides material properties and is especially useful for surface of high specularity, such as glass, metal, ceramics. Thirdly, the algorithm is designed by reversible jump and diffusion Markov chain dynamics and thus achieves globally optimal solutions under the Bayesian statistical framework. Thus it realizes the cue integration and multiple model switching. Fourthly, it adopts two techniques to improve the speed of the Markov chain search: One is a coarse-to-fine strategy and the other are data driven techniques such as edge detection and clustering. The data driven methods provide important information for narrowing the search spaces in a probabilistic fashion. We apply the algorithm to two data sets and the experiments demonstrate robust and satisfactory results on both. Based on the segmentation results, we extend the reconstruction of surfaces behind occlusions to fill in the occluded parts.

1 Introduction

Recently there are renewed and growing interest in computer vision research for parsing and reconstructing 3D scenes from range images, driven by new developments in sensor technologies and new demands in applications. Firstly, high precision laser range cameras are becoming accessible to many users, which makes it possible to acquire complex real world scenes like those displayed in Fig. 1. There are also high precision 3D Lidar images for terrain maps and city scenes with up to centimeter accuracy. Secondly, there are new applications in graphics and visualization, such as image based randering and augmented reality, and in spatial information management, such as constructing spatial temporal databases of 3D urban and suburban maps. All these request the reconstruction of complex 3D scenes, for which range data are much more accurate than other depth cues such as shading and stereo. Thirdly, Range data are also needed in studying the statistics of natural scenes[8] for the purposes of learning realistic prior models for real world imagery as well as for understanding the ecologic influences of the environment on biologic vision systems. For example, a prior model of 3D scenes is useful for many 3D reconstruction methods, such as multi-view stereo, space carving,shape recovery from occlusion, and so on.



A): A reflectance map of office A

B): A reflectance map of office B

C): A reflectance map of street scene

D): A reflectance map of cemetery

Fig. 1. Examples of indoor and outdoor scenes from the Brown dataset. A, B, C and D are reflectance images $\mathbf{I}_\Lambda$ on a rectangular lattice Λ. The laser range finder scans the scene in a cylindric coordinates and produces panoramic views of the scenes.

In contrast to the new development and applications, current range image segmentation algorithms are mostly motivated by traditional applications in recognizing industry parts in an assembly line. Therefore these algorithms only deal with polyhedra scenes. In the literature, algorithms for segmenting intensity images are introduced or extended to rang image segmentation, e.g., edge detection[9], region growing methods[5, 1] and clustering[6, 4]. An empirical comparison study was reported jointly by a few groups in [7]. Generally speaking, algorithms on range segmentation are not as advanced as intensity image segmentation, perhaps due to the lack of complex range datasets before. For example, there is no existing algorithm in the literature which can automatically segment complex scenes as those displayed in Fig. 1.

In this paper, we present a stochastic algorithm by effective Markov chain Monte Carlo (MCMC) for segmenting and reconstructing 3D scenes from laser range images and their associated reflectance maps. In comparison to previous work on range image segmentation, the paper makes the following contributions.

Firstly, to deal with the variety of objects in real world scenes, the algorithm introduces multiple types of surface models, such as planes, and conics for manmade objects, and splines for free form objects. These surfaces models compete to explain the range data under some model complexity constraints. The algorithm also introduces various prior models on surfaces, boundaries, and vertices (corners) to achieve robust solutions from noisy data.

Secondly, the algorithm integrates the range data with the associated reflectance map. The reflectance measures the proportion of laser energy returned from surface in $[0, 1]$ and therefore carries material properties. It is especially useful for surface of high specularity, for example, glass, metal, ceramics, and crucial for surface at infinity, such as the sky, where no laser ray returns. The range data and reflectance map are tightly coupled and integrated under the Bayes framework.

Thirdly, the algorithm achieves globally optimal solutions in the sense of maximizing a Bayesian posterior probability. As the posterior probability is distributed over subspaces of various dimensions due to the unknown number of objects and their types of surface models, ergodic Markov chains are designed to explore the solution space. The Markov chain consists of reversible jumps and stochastic diffusions. The jumps realizes split and merge, model switching, and the diffusion realizes boundary evolution and competition and model adaptation.

Fourthly, it adopts two techniques to improve the speed of the Markov chain search. One is a coarse-to-fine strategy which starts with segmenting large surfaces, such as sky, walls, ground, and then proceed to objects of medium sizes, such as furnitures, people etc, and then small objects such as cups, books, etc. The other are data driven techniques such as edge detection and clustering. The data driven methods provide heuristic important information, expressed as *importance proposal probabilities* [13], on the surface and boundaries for narrowing the search spaces in a probabilistic fashion.

We apply the algorithm to two datasets. The first is a standard USF polyhedra data for comparison, and the second is from Brown university which contains real world scenes. The experiments demonstrate robust and satisfactory results. Based on the segmentation results, we extend the reconstruction of surfaces behind occlusions to fill in the occluded parts.

The paper is organized as follows. We start with a Bayesian formulation in section (2). Then we discuss the algorithm in Section (3). Section (4) shows the experiments, and Section (5) discuss some problems and future work.

2 Bayes framework: integrating cues, models and priors

In this section, we formulate the problem under the Bayes framework which integrated two cues, four families of surface models, and various prior models.

2.1 Problem formulation

We denote an image lattice by

$$\Lambda = \{(i, j) : \quad 0 \leq i \leq M, 0 \leq j \leq N\}.$$

Then two cues are available. One is the 3D range data which is a mapping from lattice Λ to a 3D point,

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

(i,j) indexes a laser ray that is reflected from a surface point (x,y,z). Associated with the range data is a reflectance map

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

$\mathbf{I}(i,j)$ is the portion of laser energy returned from point $\mathbf{D}(i,j)$. G is the total grey levels in discretization. Thus it measures some material properties. For example, surfaces of high specularities, such as glass, ceramics, medals, appear dark in $\mathbf{I}$. $\mathbf{I}(i,j) = 0$ for mirrors and surfaces at infinity, such as the sky. $D(i,j)$ is generally very noisy and thus unreliable when $\mathbf{I}(i,j)$ is low, and is considered a missing point if $\mathbf{I}(i,j) = 0$. In other words, $\mathbf{I}$ and $\mathbf{D}$ are coupled at such places.

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

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

At each region R, the range data $\mathbf{D}_R$ fit to some surface model with parameter Θ^D and the the reflectance $\mathbf{I}_R$ fit to some reflectance model with parameter Θ^I.

Let W denote a solution, then

$$W = (K, \{R_n, (\ell_n^D, \ell_n^I), (\Theta_n^D, \Theta_n^I) : n = 1, 2..., K\}).$$

ℓ^D and ℓ^I index the types of surface models and reflectance models.

In the Bayesian framework, an optimal solution is searched for maximizing a posterior probability over a solution space $\Omega \ni W$,

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

In practice, two regions R_i, R_j may share the same surface model, i.e. $\Theta_i^D = \Theta_j^D$ and $\Theta_i^I \neq \Theta_j^I$. For example, a painting or a piece of cloth hung on a wall, a thin book or paper on a desk, may fit to the same surfaces as the wall or desk, but they have different reflectances. It is also possible that $\Theta_i^D \neq \Theta_j^D$ and $\Theta_i^I = \Theta_j^I$. To minimize the coding length and to pool information from pixels over more regions, we allow adjacent regions to share either depth or reflectance parameters. Thus a boundary between two regions could be labelled as *reflectance boundary*, *depth boundary*, or both.

In the following, we briefly summarize the models for $p((\mathbf{D}, \mathbf{I})|W)$ and $p(W)$

2.2 Likelihood with multiple surface and reflectance models

In the literature, there are many ways for representing a surface, such as implicit polynomials [1, 5], superquadrics [12] and other deformable models. In this paper, we choose four types of surface models to account for various shapes in natural scenes.

1. Family D_1: planar surfaces specified by three parameters $\Theta = (a, b, d)$. Let (a, b, c) be a unit surface normal with $a^2 + b^2 + c^2 = 1$, and d is the perpendicular distance from the origin to the plane. We denote by $\Omega_1^D \ni \Theta$ as the space of all planes.
2. Family D_2: B-spline surfaces with 4 control points. Each B-spline surface has a rectangular grid on reference plane ρ with its two dimensions indexed by (u, v). Then a grid of $h \times w$ control points are chosen on the ρ plane. Then a 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 B is specified by 9 parameters $\Theta = (a, b, d, \delta, \phi, \xi_{0,0}, \xi_{0,1}, \xi_{1,0}, \xi_{1,1})$. We denote by $\Omega_2^D \ni \Theta$ the family D_2.
3. Family D_3: similarly we denote by Ω_3^D the space for spline surfaces with 9 control points. So a surface in D_3 is specified by 14 parameters $\Theta = (a, b, d, \delta, \phi, \xi_{0,0}, ..., \xi_{2,2})$.
4. Family D_4: This is a surface model taken from [11] to fit spheres, cylinders, cones, and tori. D_4 is specified by 7 parameters $\Theta = (\varrho, \varphi, \vartheta, k, s, \sigma, \tau)$. We denote it by $\Omega_4^D \ni \Theta$ the space.
5. Family D_5: This is a non-parametric 3D histogram for the 3D points position. It is specified by $\Theta = (h_x 1, h_x 2, ..., h_x L_x) \times (h_y 1, h_y 2, ..., h_y L_y) \times (h_z 1, h_z 2, ..., h_z L_z)$, in which $L_x,\ L_y\ and\ L_z$ are the number of bins on x, y, z axis respectively. This model is used to represent cluttered regions, such as leaves of trees.

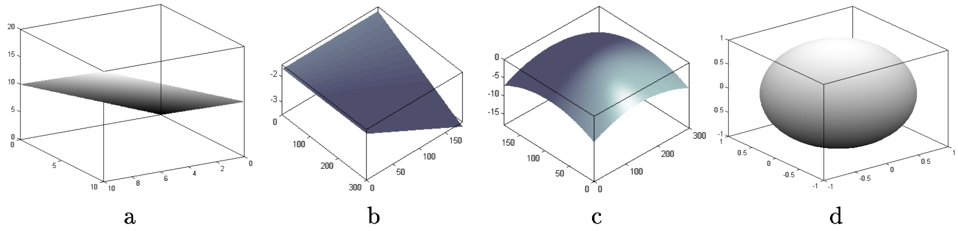


Fig. 2. One typical planar surface in family D_1, two typical B-spline surfaces in family D_2 and D_3 and one typical surface in family D_4 respectively.

Fig 2 displays four typical surfaces, one for each family.

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

1. Family $\mathbf{I}_1$: This is a uniform region with constant reflectance $\Theta = \mu$.

2. Family $\mathbf{I}_2$: This is a cluttered region with a non-parametric histogram $\Theta = (h_1, h_2, ..., h_L)$ for its intensity with L being the number of bins.
3. Family $\mathbf{I}_3$: This is a region with smooth variation of reflectance, modeled by a B-spline model as in family D_3.

For the surface and reflectance models above, the likelihood model for a solution W assumes the fitting residues to be Gaussian noise subject to some robust statistics treatment, so we have

$$p((\mathbf{D}, \mathbf{I})|W) \propto \prod_{n=1}^{K} \prod_{(i,j)\in R_n} e^{-\phi(\mathbf{D}(i,j)-S(i,j;\ell_n^D,\Theta_n^D))\delta_{\mathbf{I}(i,j)\geq\tau)}-\phi(\mathbf{I}(i,j)-J(i,j;\ell_n^I,\Theta_n^I))}.$$

In the above formula, $\phi(x)$ is a quadratic function with two flat tails to account for outliers [2] that has been used in the paper. $S(i, j; \ell_n^D, \Theta_n^D)$ and $J(i, j; \ell_n^I, \Theta_n^I)$ are respectively the fitted surface and reflectance according to model (ℓ_n^D, Θ_n^D) and (ℓ_n^I, Θ_n^I). The depth data $D(i, j)$ is not accounted for if the reflectance $\mathbf{I}(i, j)$ is lower than a threshold τ, i.e $\delta(\mathbf{I}(i, j) \leq \tau)) = 0$.

To fit the models robustly, we adopt a two step procedure. Firstly, truncate points that are less than 25% of the maximum error; Secondly, truncate points a trough or plateau. Furthermore, the least median of squares method based on orthogonal distance in [16] has been adopted for the parameters estimation.

2.3 Priors on surfaces, boundaries and corners

Generally speaking, the prior model $p(W)$ should penalize model complexity, enforce stiffness of surfaces, and enhance smoothness of the boundaries, and form canonical corners.

The prior model for the solution W is

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

where $\pi_K = (R_1, ..., R_K)$ is a K-partition of the lattice. Equally, a partition π_K is represented by a planar graph with K faces for the regions, a number of edges for boundaries, and vertices for corners,

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

Therefore, $p(\pi_K) = \prod_{k=1}^{K} p(R_K) \prod_{m=1}^{M} p(\Gamma_m) \prod_{n=1}^{N} p(V_n)$. We find that some previous prior model used by Leclerc and Fischler [10] for computing 3D wireframe from line drawings is very relevant to our prior model.

1. *Model complexity* is penalized by three factors. One is $p(K) \propto e^{-\lambda_o K}$ to penalize the number of regions, and the second includes $p(\ell_n^D)$ and $p(\ell_n^I)$ which prefer simple model. In general, for a model of type ℓ, $p(\ell)$ is proportional to the inverse of the space volume, $p(\ell) \propto 1/|\Omega_\ell|$. The third is $p(R_n) \propto e^{-\alpha|R_n|^c}$ with $|R_n|$ being the area (size) of R_n. The third term forces small regions to merge.

2. *Surface stiffness* is enforced by $p(\Theta_n^D|\ell_n^D), n = 1, 2, \cdots, K$. Every three adjacent control points in the B-spline form a plane, and adjacent planes are forced to have similar normals in $p(\Theta_n^D|\ell_n^D)$.

3. *Boundary smoothness* is enhanced by $p(\Gamma_m), m = 1, 2, \cdots, M$, like the SNAKE model. $p(\Gamma(s)) \propto e^{\int \phi(\dot{\Gamma}(s)) + \phi(\ddot{\Gamma}(s)) ds}$.

4. *Canonical corners* is imposed by $p(V_n), n = 1, 2, \cdots, N$. As in [10] the angles at a corner should be more or less equal.

3 Computing the global optimal solution by DDMCMC

Based on the previous formulation, we see that the solution space $\Omega \ni W$ contains many subspaces of varying dimensions. The space structure is typical, and will be valid even when other classes of models are used in future. In the literature of range segmentation, many methods are applied, like edge detection [9], region growing methods [5, **?**,1], clustering [6, 4], and some energy minimization method, generalized Hough transforms. But none of these methods can search in such complex spaces.

To compute a globally optimal solution, we designing ergodic Markov chains with reversible jumps and stochastic diffusions, following the successful work on intensity segmentation by a scheme called data driven Markov chain Monte Carlo [13]. The five types of MCMC dynamics used are: diffusion of region boundary, splitting of a region into two, merging two regions into one, switching the family of models and model adaptation for a region.

In a diffusion step, we run region competition [17] for a contour segment Γ_{ij} between two regions R_i and R_j. The statistical force sums over both the log-likelihood ratio of the two cues:

$$\frac{d\Gamma_{ij}(s)}{dt} = c\kappa(s)\boldsymbol{n}(s) + \log \frac{p(D(x(s), y(s)); (l_i^D, \theta_i^D))}{p(D(x(s), y(s)); (l_j^D, \theta_j^D))} + \log \frac{p(I(x(s), y(s)); (l_i^I, \theta_i^I))}{p(I(x(s), y(s)); (l_i^I, \theta_i^I))}.$$

Since two cues are used here, each region is different from its neighboring regions by either range cue or reflectance cue or both of them. Thus, in the splitting process, we can make the proposal about how to split one surface based on either the range cue or the reflectance one. This adds an extra step before splitting to select which cue will be used. As to the rest dynamics, they run in almost the similar way as in [13] with minor changes.

Now we come to the two techniques used to speed up the Markov chain search.

3.1 Coarse-to-fine Boundary Detection

In this section, we detect potential edges based on local window information of the range cue, and trace the edges to form a partition of the lattice. We organize the edge maps in three scales according to their significances. For example, Fig. 3 shows the edge maps for the office B displayed in Fig. 1. These edge maps are

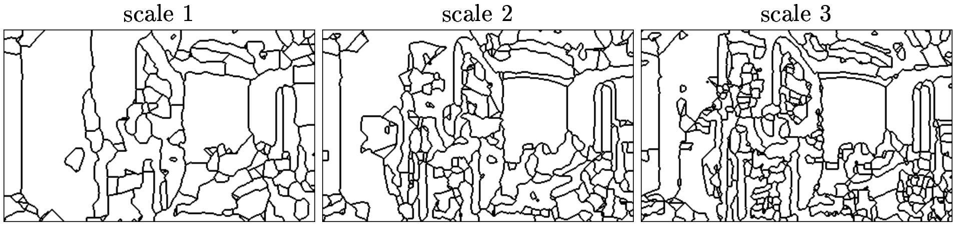


Fig. 3. Computed edge maps based on the range cue at three scales for the office scene B in Fig. 1.

used at random to suggest possible boundary Γ for the split and merge moves. Indeed, the edge maps encode some ***importance proposal probabilities*** for the jumps. Not using such edge maps is equivalent to use a uniform distribution for the edges, which obviously is inefficient. Refer to [13] for the details of how edge maps are used in designing jump. Here we deliberate on how the edge maps are computed.

At each local window, say a 5×5 patch $\omega \subset \Lambda$, we have a set of 3D points $p_i = D(m,n)$ for $(m,n) \in \omega$. One can estimate the normal of this patch by computing a 3×3 scatter matrix S [4].

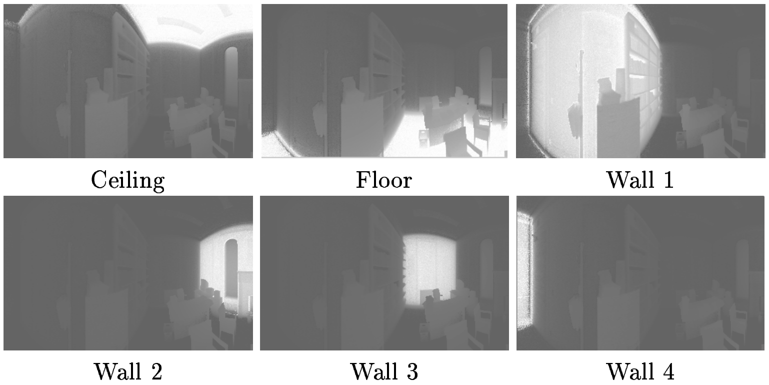
$$S = \sum_i (p_i - \bar{p})(p_i - \bar{p})^T.$$

Then the eigen vector $n = (a,b,c)$ that corresponds to the smallest eigen value λ_{min} of S is the normal of this patch ω. λ_{min} is a measure of how smooth the patch is. A small λ_{min} means a planar patch. The distance of the origin to this patch is the inner product between the unit normal and the center point $\bar{p}$, that is, $d = n \cdot \bar{p}$.

Then an edge is detected by discontinuity of (a,b,d) in adjacent pixels using standard technique, and each point is associated with an edge strength measure. We threshold the edge strength at three levels to generate edge maps which are traced with heuristic local information. We also apply standard edge detection to the refelectance image and obtain edge maps on three scales. These edge maps from the two cues are not reliable results by themselves for segmentation, but they provide important heuristic information for driving jumps of Markov chains.

3.2 Coarse-to-fine Surface Clustering

As the edge maps encode importance proposal probabilities for boundary, we compute importance proposal probabilities on the parameter spaces Ω_A, Ω_B, Ω_C and Ω_D respectively.



Ceiling Floor Wall 1

Wall 2 Wall 3 Wall 4

Fig. 4. Computed saliency maps for six clusters of office B at a coarse scale, which fit surfaces of big areas.

In edge detection, each small patch $\omega \subset \Lambda$ is characterized by $(a, b, d, \bar{p}, \lambda_0)$. Therefore, we collect a set of patches,

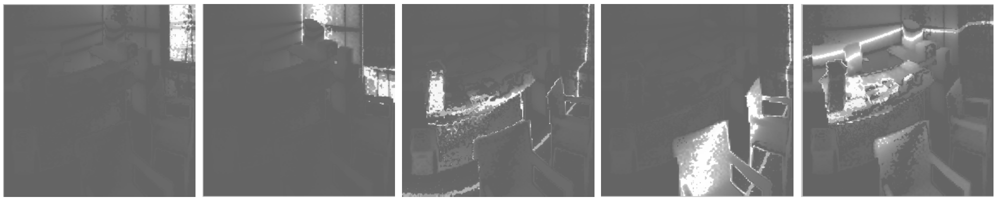
$$Q = \{\omega_j : j = 1, 2, ..., J\}.$$

In practice, we can discard patches which have relatively large λ_0, i.e. patches that are likely on the boundary. We also use adaptive patch sizes. We cluster the patches in set Q into a set of C hypothetic surfaces

$$\mathcal{C} = \{\Theta_i : \Theta_i \in \Omega_A \cup \Omega_B \cup \Omega_C \cup \Omega_D, i = 1, ..., C.\}$$

The number of hypothetic clusters in each space is chosen to be a conservative number and the clusters are computed in a coarse-to-fine strategy. That is, we extract clusters of large "populations" which usually correspond to large objects. Then we compute clusters for smaller objects. It is straight forward to do an EM algorithm that classifies the patches in S and also compute the clusters in $\mathcal{C}$. Thus we obtain a probability for how likely a patch ω_j belongs to a surface with parameter Θ_i, we denote it by $q(\Theta_i|\omega_j)$ and $\sum_i q(\Theta_i|\omega_j) = 1, \forall j = 1, ..., J$. We call $q(\Theta_i|\omega_j)$ for all $i = 1, ..., J$ the *saliency map* - a term often used by psychophysicists. A saliency map of a hypothetic surface Θ_i tells how well pixels on the lattice fit (or belong) to the surface. For example, Fig. 4 shows six saliency maps for six most prominent clusters in the office view B of Fig. 1. A bright pixel means high probability. The total sum of the probability over the lattice is a measure of how prominent a cluster is.

In our experiments, large surfaces are clustered first, as it is indicated in Fig. 4, the six most prominent clusters represents the ceiling, floor and four walls of the office scene. Other small objects, such as tables, and books do not fit well. As Fig 4 shows, many small objects do not light up in the saliency map. Thus in a coarse to fine strategy, we fit such patches at finer scale. Fig 5 shows the saliency maps of six additional clusters for a small part of the scene $\Lambda_o \subset \Lambda$. This patch is taken from the office B scene in Fig 1.



window background a PC box on desk chair backs books on desk

Fig. 5. Computed saliency maps for five clusters in a finer scale for some smaller surfaces which do not light up in the coarse scale (above). the saliency maps are shown in a zoom-in view of a patch in the office B picture.

4 Experiments

4.1 The datasets and preprocessing

We test the algorithm on two datasets. The first one is the standard Perceptron LADAR camera images in the USF dataset. These images contains polyhedra objects and the objects have almost uniform sizes. The second one is Brown dataset, where images are collected with a 3D imaging sensor LMS-Z210 by Riegl. The field of view is 80^0 vertically and 259^0 horizontally. Each image contains 444×1440 measurements with an angular separation of 0.18 degree.

In general, range data are contaminated by heavy noise. Effective preprocessing must be used to deal with all types of errors presented in the data acquisition, while preserving the true discontinuities. In our experiments, we adopt the least median of squares (LMedS) and anisotropic diffusion [14] to preprocess the range data. 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. After that, the anisotropic diffusion is adopted to handle the general noises while avoiding the side effects caused by simple Gaussian or diffusion smoothing, like the decreasing of the absolute value of curvature and smoothing of orientation discontinuities into spurious curved patch. Fig. 6 shows a surface rendered before and after the preprocessing.

4.2 Results and evaluation

Our results on the Florida dataset are shown in Fig. 7. We also show the 3D scenes by OpenGL. These 3D scenes are reconstructed by completing the surfaces behind the occluding objects. For comparison, we also show in Fig. 7 a manual segmentation used in [7]. It is no surprise that the algorithm can parse such scene very well, because the image models are sufficient to account for the surfaces in the dataset.

The results on the Brown dataset are shown in Fig. 8 and Fig. 9 respectively. Fig. 8 shows the segmentation results of two parts from office scene A and two parts from the outdoor scene C and D in Fig. 1 respectively. Fig. 9 shows the

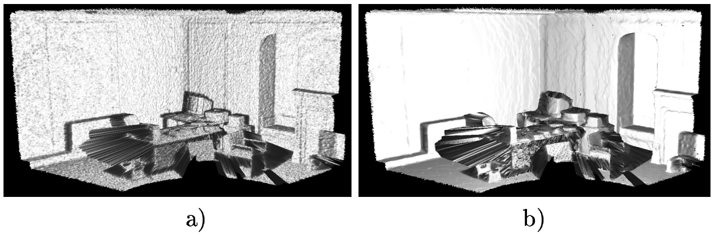


Fig. 6. Surface rendered for a range scene by OpenGL. a) before and b) after preprocessing.

segmentation result of the most complicated part of office scene B in Fig. 1. Since there is no segmentation benchmark available for such kind of complex scenes we used here, we ask several candidates to manually segment these scenes and show the average results in the above figures for comparison. The reconstruction scene based on the segmentation result shown in Fig. 9 is presented in Fig. 10.

Range image is often incomplete due to partial occlusion or poor surface reflectance. This can be clearly seen from Fig. 6, in which the floor and the two walls have a lot of missing points. Analysis and reconstruction of range images usually focuses on complex objects completely contained in the field of view; little attention has been devoted so far to the reconstruction of simply-shaped wide areas like parts of wall hidden behind furniture and facility pieces in the indoor scene shown in Fig. 6 [3]. In the reconstructing process, how to fill the missing data points of surfaces behind occlusions is a chanlenging question. The completion of these depth information needs higher level understanding of the 3D models, for examples, expressed in terms of meaningful parts. To solve this problem, an algorithm also shall make inference about two things: 1). The types of boundaries as crease, occluding, and so on. 2). The ownership of the boundary to a surface. In our reconstruction procedure, we only use a simple prior model to recover the missing parts of the backgrounds (like the walls and the floor) by assuming they are rectangles. Since we can obtain the needed parameters to represent these rectangles from the segmentation result, it is not difficult to fill the missing points. Although this procedure is pretty simple, it illustrates that our segmentation result provides a solid foundation for continuing work in this direction. Moreover, our segmentation result can also be directly applied to scene editing, which is illustrated in Fig. 10.

5 Future work

The experiments reveals the difficulty to segment degenerated regions which are essentially one-dimensional, such as the cables and rail in doorway (Fig. 8). Thus, some specific model will be integrated in future work to handle them. We should also study a more principled and systematic way for grouping surfaces into objects and completing surfaces behind the occluding objects.

Acknowledgements

This work is supported partially by two NSF grants IIS 98-77-127 and IIS-00-92-664, and an ONR grant N000140-110-535.

References

1. P.J. Besl and R.C. Jain, “Segmentation through variable order surface fitting”, *IEEE Trans. on PAMI*, vol. 10, no.2, pp167-192, 1988.
2. 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. of Comp. Vis.*, Vol. 19, No. 1 pp 57-91. 1996.
3. F. Dell'Acqua and R. Fisher, ”Reconstruction of planar surfaces behind occlusions in range images”, *To appear in IEEE Trans. on PAMI* 2001.
4. P. J. Flynn and A.K. Jain, “Surface classification: hypothesis testing and parameter estimation”, *Proc. of CVPR*, 1988.
5. A. Gupta, A. Leonardis, and R. Bajcsy, “Segmentation of range images as the search for geometric parametric models”, *Int'l J. of Computer Vision*, 14(3), pp 253-277, 1995.
6. R. L. Hoffman and A. K. Jain, “Segmentation and classification of range images”, *IEEE Trans. on PAMI*, vol. 9, no.5, pp608-620, 1987.
7. A. Hoover, et al. “An experimental comparison of range image segmentation algorithms”, *IEEE Trans. on PAMI*, vol.18, no.7, pp673-689, 1996.
8. J.G. Huang, A. B. Lee, and D.B. Mumford, “Statistics of range images”, *Proc. of CVPR*, Hilton Head, South Carolina, 2000.
9. R. Krishnapuram and S. Gupta, “Morphological methods for detection and classification of edges in range images”, *Mathematical Imaging ans Vision*, vol.2 pp351-375, 1992.
10. 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. of Comp. Vis.*, 9:2, 113-136, 1992.
11. D. Marshal, G. Lukacs and R. Martin, ”Robust segmentation of primitives from range data in the presentce of geometric degeneracy”, *IEEE Trans. on PAMI*, Vol. 23, No. 3 pp 304-314. 2001.
12. A. P. Pentland, “Perceptual organization and the representation of natural form”, *Artificial Intelligence*, 28, 293-331, 1986.
13. Z. W. Tu, S. C. Zhu and H. Y. Shum, “Image segmentation by data driven Markov chain Monte Carlo.” *Proc. of ICCV*, Vancouver, 2001.
14. M. Umasuthan and A.M. Wallace, ”Outlier removal and discontinuity preserving smoothing of range data”, *IEE Proc.-Vis. Image Signal Process*, Vol. 143, No. 3, 1996.
15. A. L. Yuille and J. J. Clark, ”Bayesian models, deformable templates and comptitive priors”, in em Spatial Vision in Humans and Robots, L. Harris and M. Jenkin (eds.), Cambridge Univ. Press, 1993.
16. Z.Y. Zhang, ”Parameter estimation techniques: A tutorial with application to conic fitting”, *Technical Report of INRIA*, 1995.
17. S. C. Zhu and A. L. Yuille. “Region competition: unifying snakes, region growing, and Bayes/MDL for multiband Image Segmentation”. *IEEE Trans. PAMI.* vol. 18, No. 9. pp 884-900. 1996.

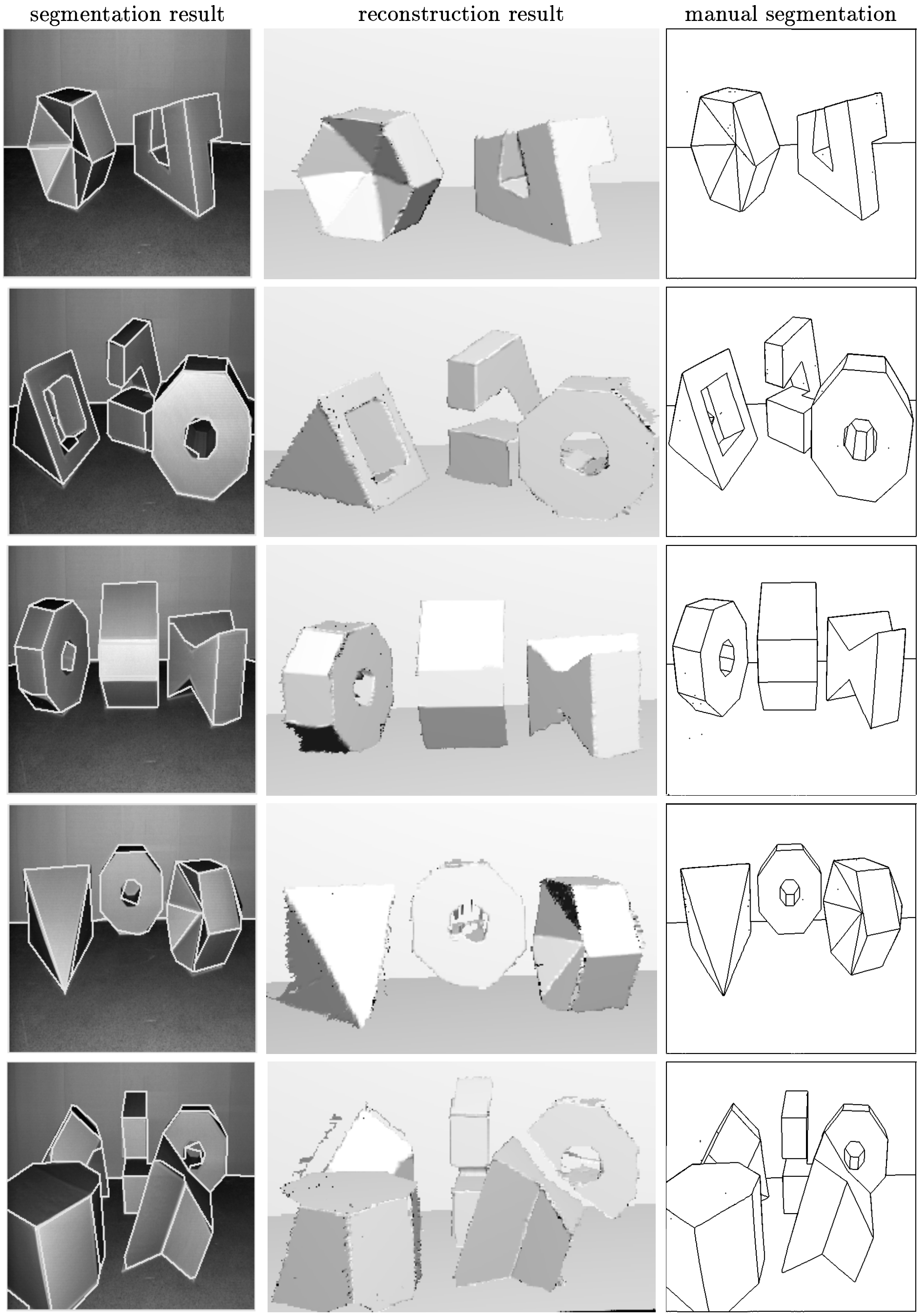


Fig. 7. Our segmentation and reconstruction results and the corresponding manual segmentation from [7] of Florida data set.

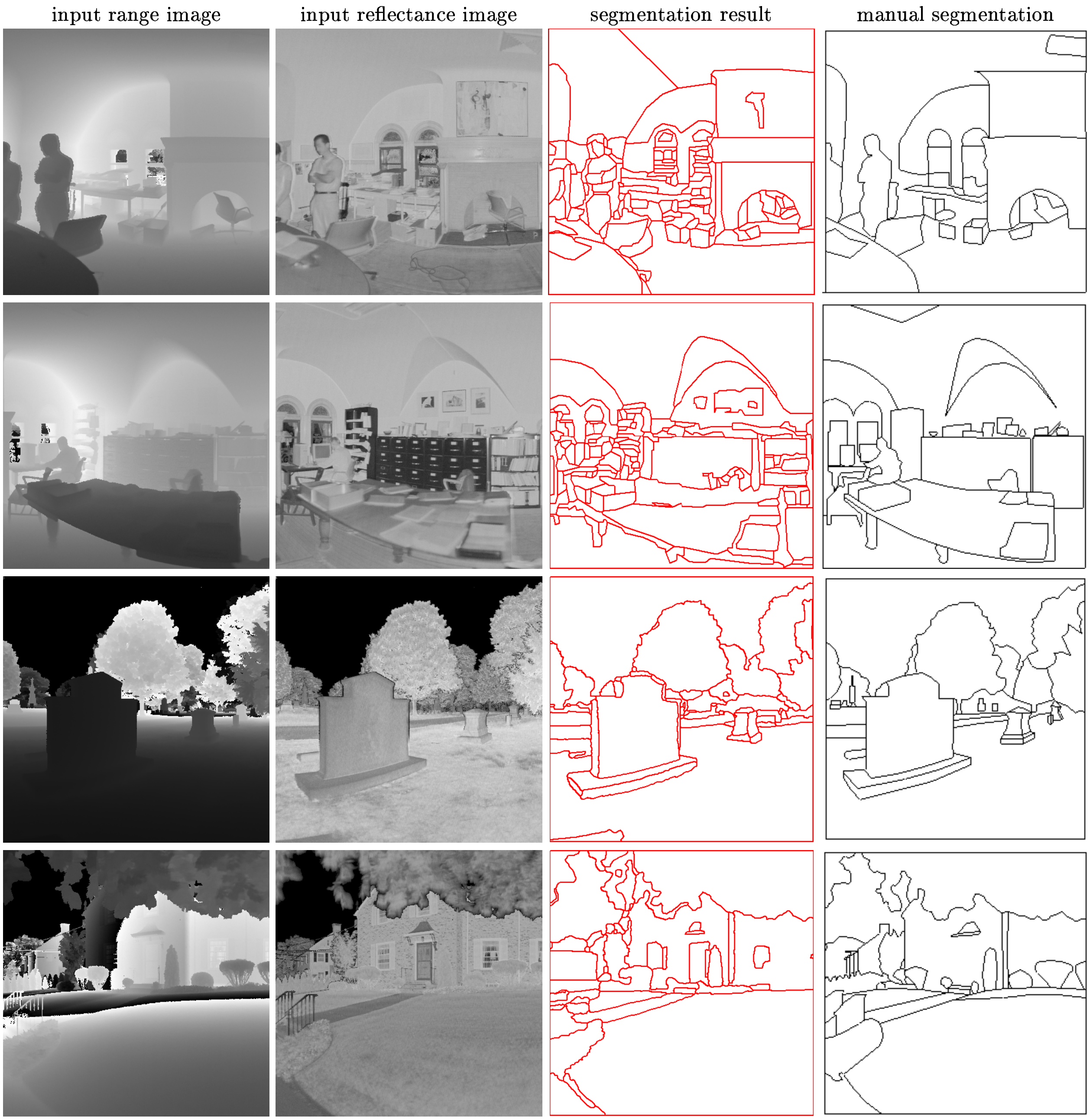


Fig. 8. Segmentation results for two parts of scene A and one part of scene C,D respectively in Fig. 1 and the corresponding manual segmentation.

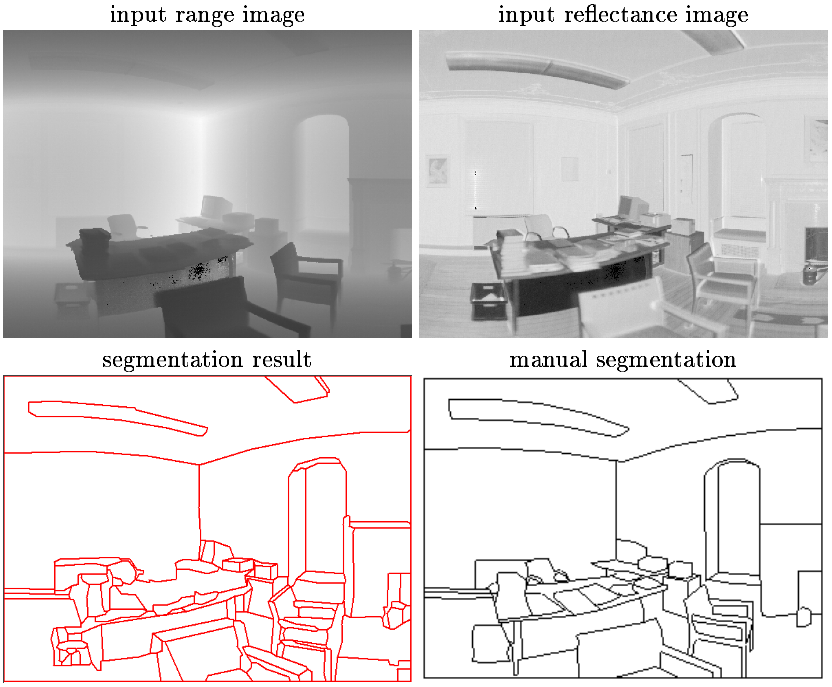


Fig. 9. Segmentation result of the most complicated part of scene B in Fig. 1 and the corresponding manual segmentation.

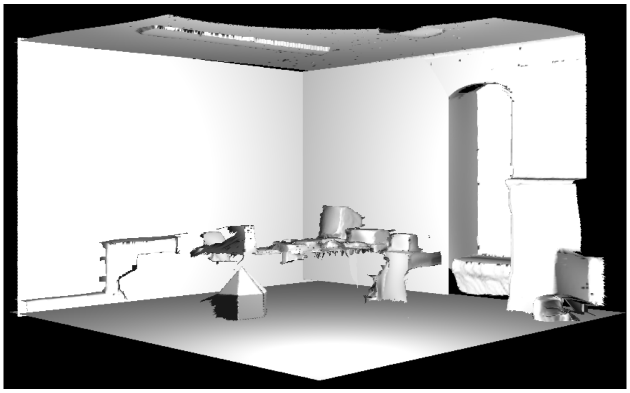


Fig. 10. The reconstructed scene based on the segmentation result shown in Fig. 9. To give the user a better view, we remove some furnitures around the table in the reconstructed scene. We also add a polyhedral object extracted from the Florida dataset to show that our segmentation result can be easily used in scene editing.