Towards Real-Time Segmentation of 3D Point Cloud Data into Local Planar Regions

Aparna Tatavarti, John Papadakis, and Andrew R. Willis
Department of Electrical and Computer Engineering
University of North Carolina at Charlotte
Charlotte, NC 28223-0001
Email: arwillis@uncc.edu

***Abstract*—This article describes an algorithm for efficient segmentation of point cloud data into local planar surface regions. This is a problem of generic interest to researchers in the computer graphics, computer vision, artificial intelligence and robotics community where it plays an important role in applications such as object recognition, mapping, navigation and conversion from point clouds representations to 3D surface models. Prior work on the subject is either computationally burdensome, precluding real time applications such as robotic navigation and mapping, prone to error for noisy measurements commonly found at long range or requires availability of co-registered color imagery. The approach we describe consists of 3 steps: (1) detect a set of candidate planar surfaces, (2) cluster the planar surfaces merging redundant plane models, and (3) segment the point clouds by imposing a Markov Random Field (MRF) on the data and planar models and computing the Maximum A-Posteriori (MAP) of the segmentation labels using Bayesian Belief Propagation (BBP). In contrast to prior work which relies on color information for geometric segmentation, our implementation performs detection, clustering and estimation *using only geometric data.* Novelty is found in the fast clustering technique and new MRF clique potentials that are heretofore unexplored in the literature. The clustering procedure removes redundant detections of planes in the scene prior to segmentation using BBP optimization of the MRF to improve performance. The MRF clique potentials dynamically change to encourage distinct labels across depth discontinuities. These modifications provide improved segmentations for geometry-only depth images while simultaneously controlling the computational cost. Algorithm parameters are tunable to enable researchers to strike a compromise between segmentation detail and computational performance. Experimental results apply the algorithm to depth images from the NYU depth dataset which indicate that the algorithm can accurately extract large planar surfaces from depth sensor data.**

I. Introduction

This article describes an approach to partition a dense collection of 3D surface measurements from an RGBD sensor into regions that are locally approximated by a single plane. This problem is generally referred to as the geometric segmentation problem. Solutions to this problem play an important role in the creation of autonomous intelligent systems where these algorithms endows systems, e.g., robotic agents, with the ability to simplify and interpret the geometric scene structure. Applications of these algorithms are often found in object recognition systems [1], [2], navigation systems [3], [4], and mapping systems [5]. Other applications are found in approaches that simplify point cloud measurements into 3D polygonal models [6], [7].

978-1-5386-1539-3/17/$31.00

Our application of interest is the use of these algorithms within mobile robots. Here, point cloud data may originate from a pair of stereo cameras [8], a laser range sensor [9], or from a depth sensor [10]. Current versions of all these sensing systems produce 3D point cloud data at rates that outpace the ability of current state-of-the-art algorithms to segment these data into planar regions.

II. Prior Work

While there have been a number of researchers investigating this problem over the years the most closely related recent work is described in the following two articles [11], [12]. In both cases the authors describe methods to quickly detect surface features from the RGBD data and then use the detected features to partition the data into local planar patches.

In [12] the primary feature for segmentation is obtained by estimating the local surface orientation by surface normal estimation. The surface orientation provides an initial partitioning of the point data which is further refined by a secondary step which clusters points based on their depth. While the authors show the application of this approach for several close range scenes, mobile robots commonly encounter surfaces at the extreme of the sensors range where the point cloud noise can be large, e.g., $\pm 7cm.$ @ $6m.$ depths for RGBD sensors [13]. In such circumstances, it is unlikely that individual point normals can be relied upon for reliable segmentation of the data as prescribed by their approach.

In [11] researchers use planar surface models to approximate subsets of the 3D point cloud data. However, they simplify the problem by leveraging the color image data from the RGBD camera. Specifically they use variations in the color information as cues for regions that are locally planar. In doing so, they condition their approach upon the stability of the color camera information which can lead to incorrect results in the presence of bright illumination/strong shadows. It also precludes direct application of their approach to purely geometric depth images as are provided by LiDAR sensors and active triangulation sensors, e.g, the Structure sensor. Further, the authors of [11] define a Markov Chain Monte Carlo (MCMC) estimation procedure to achieve their

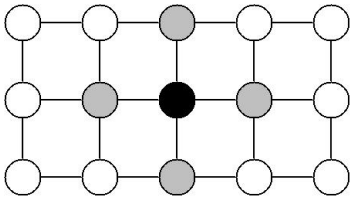


Figure 1: A site X_{ij} in a 4-connected MRF lattice is shown (black) and the neighbors of the site, $\mathcal{N}_{ij}$, are also shown (gray). The Markov property states that the distribution on X_{ij} is only a function of those random variables that are in the neighborhood $\mathcal{N}_{ij}$.

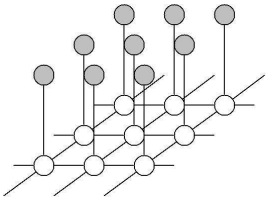


Figure 2: Energy in a MRF consists of two stochastic dependencies (shown as edges): (1) those that exist between the site labels and measured data (shown in gray) and (2) those existing between the site and neighboring site on the grid lattice (shown in white).

segmentation results and maximize the probability of their MCMC model via the Swendsen-Wang algorithm. They do not provide runtime information for this algorithm but do indicate that the performance is not real time.

In contrast to prior work, we propose an algorithm that uses planar models, a Markov Random Field and efficient Bayesian Belief Propagation to segment geometry-only depth images, i.e., *we use no color information*. We focus on extraction of large scale planes from the measured scene data and detail how our segmentation process efficiently achieves this goal. Our algorithm also includes parameters that robotics researchers can employ to strike a compromise between segmentation detail and performance.

III. Markov Random Fields

Bayes networks or, more generally, Markov Random Field (MRF) models, are probabilistic graphical models known for their ability to provide robust and accurate solutions to generic image segmentation problems [14]. For our plane segmentation application, we consider depth images, $Z(i,j)$, and proceed by imposing a MRF on the lattice defined by the image grid.

This is accomplished by defining a random variable, X_{ij}, to denote the unknown plane label for the measured 3D point at image location (i,j) and defining a distribution over this random variable that encodes the stochastic relationship between the unknown label, the observed data, $Z(i,j)$ and the labels of those points adjacent, i.e., neighboring, the point of consideration. As shown in Figure 1, we let $\mathcal{N}_{ij}$ denote the set of random variables that are members of the 4-connected neighborhood of X_{ij}. The Markov property states that the probability distribution of a site given values for the all other elements of the field is equivalent to the probability distribution obtained given only the values of those variables in the neighborhood of the site; expressed mathematically in equation (1).

$$p(X_{ij}|X - X_{ij}) = p(X_{ij}|X_{nm} \in \mathcal{N}_{ij}) \tag{1}$$

The distribution of entire field of variables, $X = \{\cup_{ij} X_{ij}\}$, represents all possible segmentations of the data and is expressed as shown in equation (2)

$$p(X) = \frac{1}{Z} e^{-U(X)} \tag{2}$$

where Z denotes the partition function that ensures the probability integrates to 1 and $U(X)$ is referred to as the energy function for the field. The energy function, $U(X)$, generally consists of two parts: (1) a *data likelihood* term, $D(Z(i,j)|X_{ij} = l)$, and (2) a *smoothness prior,* $V(X_{ij}|X_{nm} \in \mathcal{N}_{ij})$. Figure *2* graphically depicts these relations and makes clear that measured data at each grid location impacts individual sites (in gray) while stochastic relationships between site labels are shown on the lattice (in white).

MRF segmentation models seek to find the collection of label values that maximize $p(X)$. However, since $p(X)$ is an exponential distribution, the maximizer of $p(X)$ also minimizes the energy function $U(X)$. Hence, typical optimization algorithms focus on finding a global minimum of the energy function $U(X)$. Several approaches exist, but solutions to this problem impose high computational costs [15], [16]. Our optimization approach is based on work from [17] which describes an efficient method to optimize the field energy via "loopy" Bayesian Belief Propagation (BBP).

IV. Methodology

There are three steps to our approach for planar segmentation of point cloud data:

1) Detect a set of planes in the scene by fitting planar models to measured 3D point cloud data and storing those having low fit-error.
2) Cluster and merge similar planes based on their coefficients to generate a smaller set of L planes.
3) Perform BBP estimation using this reduced set of labels.

The clustering procedure removes redundant detections of planes in the scene prior to segmentation via BBP optimization of the MRF energy which can significantly reduce the BBP computational cost. MRF clique potentials dynamically change to encourage distinct labels across depth discontinuities. The net effect of these is to provide segmentation of geometry-only depth images while simultaneously controlling the computational cost. The following sections detail each step of the algorithm outlined above.

A. Planar Surface Detection

Detection of planar surfaces is accomplished by decomposing the measured 3D space into cubical regions and subsequently fitting planes to the data within these regions.

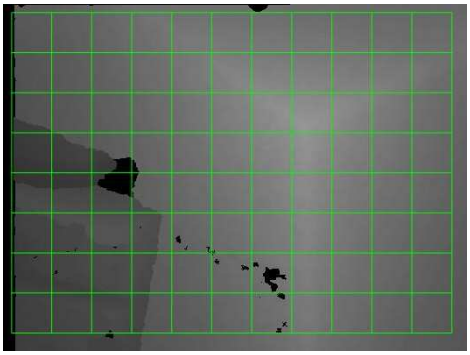


Figure 3: A tiling of an RGBD image is shown as a collection of blocks superimposed over the image. The image pixel intensities is proportional to the depth from the image plane and the green boxes denote tiles. Within each tile we fit a plane to the sensed (X, Y, Z) surface data to detect planar segments that exist in the scene. Our segmentation associates each measured point to one of the planes detected within the tiles.

In practice, the size of the cubical regions will need to be adjusted to the scale of the object being analyzed. For many sensors, e.g., stereo reconstruction, LiDAR and RGBD camera, measurements are an explicit function of the sensor location. This greatly simplifies the decomposition by tiling the field-of-view of the measurement device to partition the data. Our experiments use an RGBD sensor and tile the measured depth image as shown in Figure *3*.

Within each tile, we use the standard least-squares method to estimate the unknown plane parameters that determine the plane that minimizes the fitting error which is taken as perpendicular squared Euclidean distance between the measurements and the unknown plane. The implicit formulation seeks to minimize the square of the perpendicular distance between the measured data points and the estimated planar model, i.e.,

$$\epsilon(a,b,c,d) = \min_{a,b,c,d} \sum_{i=1}^{N} \|aX_i + bY_i + cZ_i + d\|^2 \quad (3)$$

We re-write this objective function as a quadratic matrix-vector product by defining the vector $\alpha = [\ a \quad b \quad c \quad d\ ]^t$ as the vector of planar coefficients and the matrix $\mathbf{M}$ as the matrix of planar monomials formed from the 3D (X, Y, Z) surface data having i^{th} row $\mathbf{M}_i = [\ X_i \quad Y_i \quad Z_i \quad 1\ ]$. Using this notation, the optimization function becomes:

$$\epsilon(\alpha) = \min_{\alpha} \alpha^t \mathbf{M}^t \mathbf{M} \alpha$$

As noted in several publications [18], [19], [20] the minimizer is known to be $\widehat{\alpha}$, the eigenvector associated with the smallest eigenvalue of the matrix $\mathbf{M}^t\mathbf{M}$ (also known as the scatter matrix). In general, $\mathbf{M}^t\mathbf{M}$ is a symmetric matrix and, for the monomials $\mathbf{M}_i = [\ X_i \quad Y_i \quad Z_i \quad 1\ ]$, the elements of this matrix are

$$\mathbf{M}^t\mathbf{M} = \sum_{i=1}^{N} \begin{bmatrix} X_i^2 & X_iY_i & X_iZ_i & X_i \\ X_iY_i & Y_i^2 & Y_iZ_i & Y_i \\ X_iZ_i & Y_iZ_i & Z_i^2 & Z_i \\ X_i & Y_i & Z_i & 1 \end{bmatrix} \quad (4)$$

The squared Euclidean error between the measured 3D data and the fit plane is obtained by normalizing the coefficients such that the coefficients a, b, c form a vector of unit length. Let $\eta = \sqrt{a^2 + b^2 + c^2}$ denote this normalization constant and we can then write the sum of squared Euclidean errors between the tile points and the fit surface as $\epsilon(\widehat{\frac{\alpha}{\eta}}) = \frac{\sqrt{\lambda_{11}}}{\eta}$.

For each tile, our plane detection algorithm stores the (x, y) position of the tile, the normalized parameters of the fit plane, $\widehat{\frac{\alpha}{\eta}}$, and the error observed between the plane and the data, $\epsilon(\widehat{\frac{\alpha}{\eta}})$. This generates an image of planar fits that, like the image data, is organized on a grid.

B. Merging Planes

The next step for our algorithm is to merge estimates of the same plane. This step serves to reduce the number of potential classes that must be considered during the segmentation problem. Standard implementation of BBP algorithms must compute the probability of each candidate classes at each individual pixel. Specifically, the cost of a single iteration of BBP for an $N \times M$ image having L candidate labels is $\mathcal{O}(NML^2)$ [17]. By merging the list of candidate planes we reduce potential over-segmentation of points lying on a same plane, i.e., having multiple instances of the same plane class, while simultaneously reducing the computational cost of the BBP algorithm.

Our merge procedure is a quick algorithm based on a clustering method that uses orthogonal projections of the plane coefficient data into 1-dimensional subspaces [21]. In our application, we cluster planes by projecting the coefficients of the plane models onto each of the 4 plane-parameter axes, i.e., the $a, b, c, d-$axes. In each case, we merge plane models that are adjacent on the given axis and satisfy the relation shown in equation (5).

$$\epsilon(\alpha_i, \alpha_j) = 1 - (a_ia_j + b_ib_j + c_ic_j) + \beta\,|d_i - d_j| < \upsilon \quad (5)$$

where υ denotes a similarity threshold and the plane pair (α_i, α_j) will be merged when $\epsilon(\alpha_i, \alpha_j) < \upsilon$. When two similar plane models are merged we discard the coefficients of the plane having larger fit error, $\epsilon(\widehat{\frac{\alpha}{\eta}})$, and assign the data associated with the discarded plane to reference the remaining plane model.

C. Bayesian Belief Propagation

The final step of our algorithm segments the measurement data to one of the classes from the set of L planes remaining after the merging procedure. This is accomplished by writing the segmentation as a MRF and then searching for the MAP estimate of the labels for each pixel.

In practice, the MRF energy is computed as the sum of the site energies. Hence the MRF is determined by defining the energy for site X_{ij} as shown in equation (6) and then the total field energy is obtained by summing this energy across all sites.

Our site energy function seeks to assign labels to by striking a balance between the error in the planar fit for each measured

surface point, $D(Z(i,j)|X_{ij}=l)$ and preserving uniformity, i.e., smoothness in the assigned labels, $V(X_{ij}|X_{nm}\in\mathcal{N}_{ij})$.

$$U(X_{ij}) = \sum_{l\in L} D(Z(i,j)|X_{ij}=l_i) \quad + \quad ... \\ \sum_{(nm)\in\mathcal{N}_{ij}} V(X_{ij}=l_i|X_{nm}=l_j) \tag{6}$$

Segmentation results rely heavily on the form of the data likelihood energy, $D(Z(i,j)|X_{ij}=l)$, and the label smoothness energy, $V(X_{ij}=l_i|X_{nm}=l_j)$.

Our proposed data likelihood energy uses a truncated cost function given in equation (7).

$$D(Z(i,j)|X_{ij}=l) = \lambda \min\left(|a_l X + b_l Y + c_l Z + d|, \tau\right) \tag{7}$$

The likelihood energy function encodes the stochastic relationship between measured depth $Z(i,j)$ and the unknown label value, l. Equation (7) states that the cost of associating the depth measurement $Z(i,j)$ to the plane having label, l, and plane coefficients $\alpha_l = \{a_l, b_l, c_l, d_l\}$. Note that this cost is proportional to the perpendicular Euclidean distance between the measured (X,Y,Z) point and plane α_l. The likelihood energy function includes two free parameters λ and τ. The λ parameter controls the relative weight of the terms $D(Z(i,j)|X_{ij}=l)$ and $V(X_{ij}|X_{nm}=x_{nm})$ in the total site energy $U(X_{ij})$. The τ parameter is used to restrict the range of the data energy cost which has been found to improve the robustness of MRF estimation procedures and denotes the maximum allowable cost between any given label and a measurement. Limiting the cost in this way reduces the sensitivity of the segmentation result to outliers in the measurement data.

Our proposed smoothness energy differs from typical smoothness constraints which typically rely only on the value of the labels. For example, the Potts model [22] imposes a cost penalty when the label for a given site is different than it's neighbor and zero cost otherwise as shown in equation (8).

$$V(X_{ij}=l_i|X_{nm}=l_j) = \begin{cases} \gamma & if \quad l_i \neq l_j \\ 0 & if \quad l_i = l_j \end{cases} \tag{8}$$

Since optimization seeks to minimize the cost function, this energy term "smooths" the label assignments by encouraging neighboring locations to share the same label values generating piecewise constant label regions. The cost/penalty, γ, for assigning distinct labels to neighboring MRF sites, i.e., pixel locations, is also referred to as a "discontinuity" cost.

In contrast to typical practice, our smoothness cost is composed of two parts: (1) a cost penalizing distinct neighboring label values (similar to equation (8)) and (2) a data likelihood term which serves to modulate the cost by how well the planar model predicts the observed depth difference.

For (1) we use the same plane dissimilarity metric previously applied for clustering as shown in equation (8).

$$\epsilon(\alpha_i,\alpha_j) = 1 - (a_i a_j + b_i b_j + c_i c_j) + \beta\,|d_i - d_j| \tag{9}$$

Equation (5) which has value 0 when $l_i = l_j$ (equivalently $\alpha_i = \alpha_j$) and will evaluate to a value of at least υ for all other label pairs $l_i \neq l_j$ due to the clustering stage of §IV-B. Note that these costs will increase for label pairs having large dissimilarity and is not constant.

For (2) we use the likelihood of the measured depth difference given the label value, i.e., plane model. Theoretically, we could make use of the plane coefficients to determine how well the depth difference is predicted by the plane coefficients. Consider two neighboring depth measurements Z_1 and Z_2. Hypothesized to lie on the plane $aX+bY+cZ+d=0$. If this hypothesis is true, the predicted value of the depth at Z_2 will be $\widehat{Z_2} = Z_1 + c(Z_2 - Z_1)$. This can be converted to a cost function $D(Z_1,Z_2|c) = \left|Z_2 - \widehat{Z_2}\right| = |Z_2 - Z_1 - c(Z_2 - Z_1)| = |(1-c)(Z_2 - Z_1)|$ that adds cost when the observed the depth difference is different from that predicted by hypothesized plane model having coefficient c. In practice, we discard the term in c and use the cost $D(Z_1,Z_2) = |Z_2 - Z_1|$ which approximates the theoretical analysis.

The hybrid smoothness energy term resulting from parts (1) and (2) are combined into a single smoothness energy function as shown in equation (10).

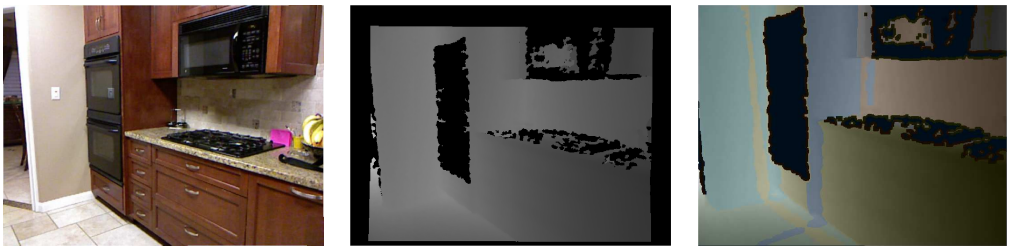
$$V(X_{ij}=l_i|X_{nm}=l_j) = \max\left(|Z(i,j) - Z(m,n)|, \epsilon(\alpha_i,\alpha_j)\right) \tag{10}$$

Where $\epsilon(\alpha_i,\alpha_j)$ is the plane dissimilarity metric of equation (9) for the plane pair (α_i,α_j) having labels (l_i,l_j) and $|Z(i,j) - Z(m,n)|$ is the absolute depth difference for the neighboring pixels. Introduction of this term improves segmentation as labels assignments that span large depth discontinuities do not incur the standard label smoothness penalties as is typical to a Potts model and other prior work on geometric segmentation [11], [12]

D. MRF Energy Optimization via BBP

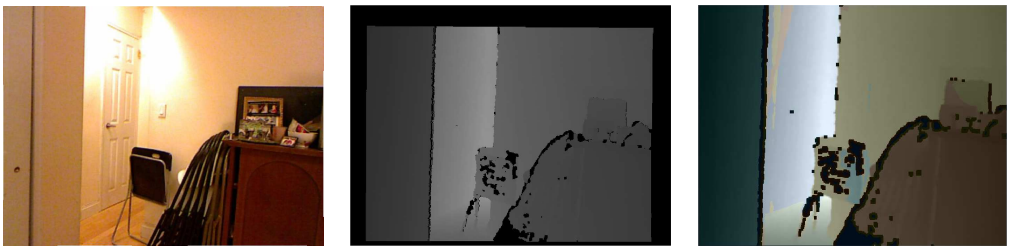
Our optimization of the MRF energy uses the Bayesian Belief Propagation (BBP) algorithm to find the MAP estimate of the segmentation label values. Computational cost is reduced by using the max-sum algorithm and passing messages using the "checkerboard" message passing scheme described in [17], which takes advantage of regularity of the image grid to formulate the loopy BBP message passing as a two-step message passing process on a bipartite graph. This alternative message passing scheme decreases both the runtime and memory requirements by a factor of two when compared to traditional BBP message passing schemes. After a number of iterations of message passing, the maximum a posteriori label assignment is evaluated.

As mentioned in [17], several important classical computer vision problems can be formulated with simple smoothness energy functions which often include a single term that depends only on the difference between label values, i.e., $V(X_{ij}=l_i|X_{nm}=l_j,\, X_{nm}\in\mathcal{N}_{ij}) = |l_i - l_j|$. This criteria is the key attribute that allows the computational complexity of an iteration of the BBP algorithm to be reduced from $\mathcal{O}(NML^2)$ to $\mathcal{O}(NML)$, where NM denotes the number



(a) (b) (c)

Figure 4: Planar segmentation algorithm applied to a kitchen area. Regions of similar color have been classified as coplanar. The algorithm performs well, segmenting the image into the planar surfaces such as walls, floors, and cabinetry.



(a) (b) (c)

Figure 5: Planar segmentation algorithm applied to a portion of a bedroom. Large scale features such as walls and the floor are well segmented.

(a) (b) (c)

Figure 6: Planar segmentation algorithm applied a hallway scene. The data belonging the walls, floors, door, and cupboard in the scene are relatively well classified.

of MRF sites/pixels in the image and L denotes the number of segmentation labels. It is important to note that this result also holds for more general MRF label smoothness functions. Specifically, a sufficient condition for this performance boost requires only that the smoothness energy function, $V(X_{ij} = l_i | X_{nm} = l_j,\ X_{nm} \in \mathcal{N}_{ij})$, increases as a monotonic linear or quadratic function of the label difference.

Hence, for our algorithm or, more generally, any BBP-based optimization to benefit from the computational gains of [17], a re-ordering of the labelset must be found that satisfies this monotonicity constraint. Unfortunately, it does not appear that, in general, such an arrangement exists for an arbitrarily large set of plane models using the cost metric of equation (10). As such, our BBP algorithm computational complexity remains $\mathcal{O}(NML^2)$ and requires two passes over the labelset to compute BBP messages rather than one. Despite this fact, our merge procedure (§IV-B) aggressively clusters planes to generate a small number of candidate plane models. Hence, the constant L is typically small in our implementation which controls the relative impact of the computational complexity. Exploitation of this attribute to improve computational efficiency is a topic of future research.

V. Results

Our experiments use depth data from the Microsoft Kinect sensor provided by the NYU RGBD Dataset [23] which include depth images from multiple indoor scenes. The RGBD sensor has a full-frame resolution of 640x480 pixels and a maximum depth range of ~6m. The BBP optimization algorithm [17] and our modifications to this algorithm as described in § IV were implemented as MATLAB programs. Figures 4,5 6, and show the results of the planar segmentation algorithm on multiple indoor scenes including a kitchen area, bedroom, and hallway. Parameters used to generate these segmentations include a blocksize 40x40 px, $\beta = 0.4$, $\upsilon = 0.15$, and 5 iterations of belief propagation with $\lambda = 0.2$, $\tau = 0.5$. For

these figures, the first column shows a color image of the scene, the middle column shows the registered depth image, and the last column shows the colored segmentation labels superimposed on the depth image.

Figure 4 shows the segmentation of a kitchen area. The algorithm performed well, separating large scale features such as walls, floor, and cabinetry. The algorithm picked up even the small surface patches below and above the oven. Parameters such as the blocksize and planar merge threshold can be adjusted to better target planar surfaces of various sizes. Figure 5 shows the segmentation of hallway. The segmentation algorithm approximated the image of hallway well with a total of 5 different planes. The larger surfaces such as walls, floor, and the cabinet were well identified and labeled as separate planes. Figure 6 shows the segmentation of bedroom. The algorithm segmented large scale planar features such as the walls, cupboard, door and floor particularly well, smaller planar surfaces such as the chair were missed due to the choice in block size. Reliable segmentation and extraction of these large scale planar features is of interest for many vision related problems.

VI. Conclusion

In this article we have described a novel algorithm to segment point cloud data into local planar regions. This is a problem of generic interest to researchers in the computer graphics, computer vision, artificial intelligence and robotics community where it plays an important role in applications such as object recognition, mapping, navigation and conversion from point clouds representations to 3D surface models. In contrast to prior work, we propose an algorithm that uses planar models, a Markov Random Field and efficient Bayesian Belief Propagation to segment geometry-only depth images, i.e., *we use no color information*. The fast clustering technique applied removes redundant plane detections prior to optimization to improve performance. A new MRF smoothness energy function dynamically changes to encourage distinct labels across depth discontinuities. These modifications provide improved segmentations for geometry-only depth images while simultaneously controlling the computational cost. We focus on extraction of large scale planes from the measured scene data and detail how our segmentation process efficiently achieves this goal. Our algorithm includes parameters that robotics researchers might employ to strike a compromise between segmentation detail and performance. Experimental results apply the algorithm to the NYU depth dataset and indicate that the algorithm can accurately segment a variety of planar surfaces from depth sensor data.

References

[1] S. Pillai and J. Leonard, "Monocular slam supported object recognition," in *Proceedings of Robotics: Science and Systems*, 2015.

[2] K. Lai, L. Bo, X. Ren, and D. Fox, "Rgb-d object recognition: Features, algorithms, and a large scale benchmark," in *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*, pp. 167–192, Springer, 2013.

[3] P. H. J. K. A. J. D. Renato F. Salas-Moreno, Ben Glocker, "Dense planar slam," in *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 157–164, 2014.

[4] K. Pathak, A. Birk, N. Vaskevicius, M. Pfingsthorn, S. Schwertfeger, and J. Poppinga, "Online 3d slam by registration of large planar surface segments and closed form pose-graph relaxation," *Journal of Field Robotics, Special Issue on 3D Mapping*, 2010.

[5] V. Nguyen, A. Harati, A. Martinelli, N. Tomatis, and B. Sa, "Orthogonal slam: a step toward lightweight indoor autonomous navigation," in *In: Proceedings of the IEEE/RSJ Intenational Conference on Intelligent Robots and Systems, IROS*, 2006.

[6] M. Pauly, M. H. Gross, and L. Kobbelt, "Efficient simplification of point-sampled surfaces," in *IEEE Visualization*, 2002.

[7] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007.

[8] F. Tombari, S. Mattoccia, and L. D. Stefano, "Stereo for robots: Quantitative evaluation of efficient and low-memory dense stereo algorithms," in *2010 11th International Conference on Control Automation Robotics Vision*, pp. 1231–1238, Dec 2010.

[9] W. Maddern and P. Newman, "Real-time probabilistic fusion of sparse 3d lidar and dense stereo," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2181–2188, 2016.

[10] G. Loianno, V. Lippiello, and B. Siciliano, "Fast localization and 3d mapping using an rgb-d sensor," in *2013 16th International Conference on Advanced Robotics (ICAR)*, pp. 1–6, Nov 2013.

[11] C. Erdogan, M. Paluri, and F. Dellaert, "Planar segmentation of rgbd images using fast linear fitting and markov chain monte carlo," in *2012 Ninth Conference on Computer and Robot Vision*, pp. 32–39, May 2012.

[12] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, *Real-Time Plane Segmentation Using RGB-D Cameras*, pp. 306–317. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[13] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, p. 1437, 2012.

[14] S. Z. Li, *Markov Random Field Modeling in Image Analysis*. Springer Publishing Company, Incorporated, 3rd ed., 2009.

[15] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, (San Francisco, CA, USA), pp. 467–475, Morgan Kaufmann Publishers Inc., 1999.

[16] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in gaussian graphical models of arbitrary topology," *Neural Comput.*, vol. 13, pp. 2173–2200, Oct. 2001.

[17] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 41–54, 2006.

[18] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, pp. 476–480, May 1999.

[19] G. Taubin, "Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1115–1138, 1991.

[20] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[21] D. Han and J. Kim, "Unsupervised simultaneous orthogonal basis clustering feature selection," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5016–5023, June 2015.

[22] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, pp. 1222–1239, Nov. 2001.

[23] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.