2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Efficient Segmentation and Surface Classification of Range Images

Georg Arbeiter, Steffen Fuchs, Joshua Hampp and Richard Bormann
Fraunhofer IPA
Stuttgart, Germany
Email: {goa, steffen.fuchs, josh, rmb}@ipa.fhg.de

***Abstract*—Derivation of geometric structures from point clouds is an important step towards scene understanding for mobile robots. In this paper, we present a novel method for segmentation and surface classification of ordered point clouds. Data from RGB-D cameras are used as input. Normal based region growing segments the cloud and point feature descriptors classify each segment. Not only planar segments can be described but also curved surfaces. In an evaluation on indoor scenes we show the performance of our approach as well as give a comparison to state of the art methods.**

I. Introduction

3-D perception is one of the crucial fields of research towards fully autonomous and robust service robot operation in the next years. Without perception of the environment, service robots will not be able to navigate, manipulate or interact with humans successfully.

Mobile service robots like the Care-O-bot 3® are designed to accomplish household tasks in unstructured environments. Thus, they are equipped with a mobile base, a manipulator and various sensors to perceive the environment.

During recent years, algorithms for processing point cloud data have evolved greatly. Starting from advanced filtering, continuing with registration and map management and finishing with segmentation, feature extraction and surface reconstruction, there is a huge portfolio of methods available. With the introduction of RGB-D cameras like *Microsoft Kinect*, applications of such methods had a major boost. Being cheap and despite having far better data quality than their predecessors, they lay a base for successful and robust 3-D perception in unstructured environments.

Derivation of geometric information from point clouds is one of the major steps in scene understanding. With the extraction of geometric primitives, the data is separated into descriptive segments. This significantly decreases the amount of data while keeping or even enriching the information. Methods for segmentation and classification are manifold. However, many approaches lack in efficient computation and ignore the ordered structure of the data coming from RGB-D cameras. Additionally, robustness towards under- or over-segmentation is often neglected. Furthermore, combination of segmentation and classification seems to be done rarely.

In this paper, we propose a novel method for segmentation and surface classification of ordered point clouds (depth images). We introduce (1) an efficient normal based region growing method that results in a graph structure followed by a (2) refinement step dealing with over-segmented regions. This results in segments containing points with common geometric properties. These properties are (3) classified by point feature descriptors in order to produce geometric primitives like planes, spheres or cylinders. The primitives are (4) finally expressed by their parameters and hull points in order to create a memory-efficient representation.

The remainder of this paper is structured as follows: In section II, we give an overview on related work. This is followed by the methodology of our approach in section III. Afterwards, experimental results on indoor scenes are presented in section IV. Finally, we give a conclusion and insights on future work in section V.

II. Related Work

Segmentation of point clouds is a well researched topic. Basically, there are two main strategies, edge based and region based segmentation. Edge based approaches try to identify closed edges in the data. Every region within a closed contour forms a segment. In [1], point normals and jumps in depth data are used to identify edges. In [2], local curvature is used to identify edges. Yang and Lee describe in [3] how to identify edges using a scan-line approach. The main challenge occurring with edge based methods is to generate closed contours from the edges. In [4], a topological analysis is carried out to build a contour graph. Hsiao et al. use morphologic operators to close edge segments [5]. A graph based strategy is used in [6], [7]. Most of the edge based approaches are computationally expensive and therefore not suitable for real-time applications. Another problem is noisy data. As it is crucial to identify closed contours for these methods, sensor noise reduces robustness significantly.

In contrast, region based approaches identify homogeneous segments in a point cloud. In *top-down* approaches, parametric models are fit to point clouds. In this case, RANSAC is the most common algorithm [8], [9], [10], [11]. Mainly planar models are used. Other approaches include color information [12], Hough transform [13] or point normals [14]. Alternatively, *bottom up* methods like region growing aggregate all points to a segment that fulfil a similarity criterion. In [15], [16], point normals are used to measure similarity, for example. Rusu, however, shows region growing based on border points in [17]. In [18], model parameters are estimated

in advance followed by a step that assigns as many points to the model as possible. An efficient method that only works for planes is shown in [19]. A similar approach called Multi Plane Segmentation (MPS) and using connected components has been published recently as part of the PCL library [20]. Region based methods seem to be applicable in a more general manner. The main challenge is to find a similarity criterion that works in all situations and different data. Otherwise, under- or over-segmentation may occur. Also, most approaches only segment planar regions. The main problem with RANSAC based approaches is that they are not model consistent (the best model fit is done even if the data represents something else). Also, the model to be searched has to be known beforehand.

If parametric model fitting is used for segmentation, the surface class is known without further processing. Having a region growing segmentation, it is necessary to classify the segment surfaces. Point feature descriptors are suitable for fulfilling this task. Three promising feature types are *Fast Point Feature Histograms* (FPFH) [21], *Principal Curvatures* (PC) [20] and *Radius-based Surface Descriptor* (RSD) [22]. All of them can distinguish between multiple surface classes like planes, edges, cylinders or spheres on a per-point base. As result of our evaluation in [23], PC is the most suitable descriptor for our purposes.

Our approach, in contrast to state of the art, focuses on efficient segmentation on ordered range images. We choose region growing with point normals as similarity criterion, similar to the approaches mentioned. However, we introduce a refinement step to cope with over-segmentation which increases robustness. Different to existing methods, our approach is able to not only segment planar surfaces but as well curved ones intrinsically. Thus, we propose surface classification after the segmentation using point feature descriptors.

III. Methodology

The algorithm proposed first estimates the point normals. Afterwards, region growing based on the normals is done in order to segment the point cloud. A graph structure is created simultaneously. This structure is used in a refinement step to reduce over-segmentation. Finally, the surface of each segment is classified using PC. All calculations exploit the ordered structure of range images in order to be efficient. The pseudocode for the algorithm is shown in Fig. 1.

A. Normal Estimation

Point normals are estimated w.r.t. the underlying local surface. To derive the surface structure, nearest neighbors $\boldsymbol{q}$ to the query point have to be found. We propose an efficient nearest neighbor search based on the pixel structure. A 2-D search mask as shown in Fig. 2 is defined. The mask consists of several frames with pixel distance s_f around the query point. On each frame, mask points are set with a distance s_p. The number of frames is defined by the desired overall search radius s_r. The mask is applied to the point cloud with the query point $\boldsymbol{p}_q$ as center. Thus, neighbor points to $\boldsymbol{p}_q$ can be located with a certain density. Furthermore, a depth threshold

Require: Ordered point cloud $P = \{\boldsymbol{p}_i\}$ with c columns and r rows
Ensure: Surface class for each segment S
for all $\boldsymbol{p}_i$ **do**
  Calculate point normal $\boldsymbol{n}_i$
end for
procedure GROW SEGMENTS(P)
  Define neighborhood of point $\mathbf{p}_{i,j}$ as $\mathcal{P}_{i,j} = \{\mathbf{p}_{i,j-1}, \mathbf{p}_{i,j+1}, \mathbf{p}_{i-1,j}, \mathbf{p}_{i+1,j}\}$
  Label l_p of point $\mathbf{p}_{i,j}$
  for $i = 1$ to c **do**
    for $j = 1$ to r **do**
      if $\nexists l_p$ **then**
        wave-front $w \leftarrow \mathbf{p}_{i,j}$
        $l_p \leftarrow$ new label
      end if
      while w is not empty **do**
        Compute average normal $\bar{\boldsymbol{n}}$
        for all $\mathbf{q} \in \mathcal{P}_{i,j}$ **do**
          if $\nexists l_q$ and c_1 and c_2 **then**
            $w \leftarrow \{w, \mathbf{q}\}$
            $l_q \leftarrow l_p$
          end if
        end for
        $\mathbf{p_{i,j}} \leftarrow w(\text{end})$
      end while
    end for
  end for
  Extend Graph G
end procedure
procedure REFINE SEGMENTATION(S,G)
  Use G to determine adjacent segments S_j, S_k
  Compute smoothness probability P_s of border point pairs $\langle \boldsymbol{b}_j, \boldsymbol{b}_k \rangle$
  if $P_s > P_{th}$ **then**
    Merge S_j, S_k
  end if
end procedure
procedure CLASSIFY(S)
  for all $S_i \in S$ **do**
    Compute PCA over point normals
    Compute PC on non-planar segments
    Apply label according to dominant class
  end for
end procedure

Fig. 1: Normal based region growing and surface classification

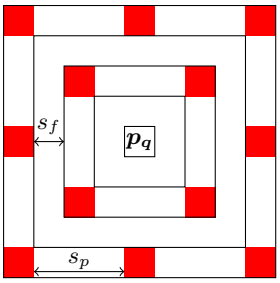


Fig. 2: The search mask.

is checked for each possible neighbor point in order to reject points with depth values far away from $\boldsymbol{p}_q$. By using a fixed mask, neighbor search timing does not scale with the number of points in the cloud which is the case for existing ANN search methods utilizing radius-based search. Our method can be considered as a local adaptive down-sampling of the range image. As the density of data from RGB-D cameras corresponds with the distance of objects to the sensor, also the down-sampling of the neighborhood is coarser in regions far away from the camera.

Once the neighbor points are found, the point normals can be efficiently calculated by

$$\boldsymbol{n}_q = \sum_{ij, i \neq j} \left(\boldsymbol{q}_i - \boldsymbol{p}_q\right) \times \left(\boldsymbol{q}_j - \boldsymbol{p}_q\right) \tag{1}$$

as sum of the vector products of the vectors between $\boldsymbol{p}_q$ and its neighbors.

B. Segmentation

For segmentation, the point cloud is processed by region growing starting in the upper left corner of the range image. The goal is to aggregate regions with similar normal vectors into segments. During growth of a region, the depth values of each new segment candidate $\boldsymbol{q}$ is compared to the direct neighbor $\boldsymbol{p}$ belonging to the segment. This leads to a first growth criterion

$$c_1 : \ \|\boldsymbol{q} - \boldsymbol{p}\| < d_{th} \tag{2}$$

with the adaptive threshold $d_{th} = c\boldsymbol{p}^2$. This accounts for the quantization noise of RGB-D cameras that scales quadratically with depth [24]. If a candidate passes the first threshold, the average normal $\bar{\boldsymbol{n}}$ is computed by

$$\bar{\boldsymbol{n}} = \frac{1}{k} \sum_k \boldsymbol{n}_k \tag{3}$$

from the k normal vectors currently belonging to the segment. The second growth criterion is

$$c_2 : \ \arccos\left(\bar{\boldsymbol{n}} \cdot \boldsymbol{n}\right) < \alpha_{th} \tag{4}$$

for the angle α between the normalized point normals $\boldsymbol{n}$ and $\bar{\boldsymbol{n}}$. If both criteria are met, the query point is added to the current wave-front w. A segment is grown until w is empty. To make segmentation more robust, the region growing first processes all neighbors with a similar α first. Thus, each segment starts to grow into homogeneous regions before reaching the borders. Furthermore, a graph structure is built during growth. Each node represents a segment and each edge defines a connecting border between segments. As can be seen in Fig. 3, for each edge, a set of neighboring border point pairs $B = \{\langle \boldsymbol{b}_j, \boldsymbol{b}_k \rangle\}$ is stored.

C. Refinement

The parameters of the segmentation step are set to values which lead to over-segmentation, especially for non-planar regions. As we propose a refinement step utilizing the graph structure, that compensates over-segmentation, this behaviour is desired. Goal of the refinement is to merge segments that belong to the same surface. For each connected segment pair S_1, S_2, the border point pairs B are observed. For each point pair $\{\langle \boldsymbol{b}_j, \boldsymbol{b}_k \rangle\}$, the point normals are re-computed using only the segment inliers in order to diminish influence of neighboring segments. Afterwards, the angle β between each normal pair is calculated in analogy to (4). For each B, the probability P_s of being a smooth border can be expressed by

$$P_s = \frac{n_s}{n_B} \tag{5}$$

with the number of border point pairs n_s with $\beta \leq \beta_{th}$ and the total number of border point pairs n_B. If P_s exceeds a threshold P_{th}, S_1 and S_2 are considered to belong to the same surface and merged.

D. Surface Classification and Parametric Description

Once the segments are refined, the surface defined by each segment can be estimated. We distinguish between planar surfaces and more complicated structures. For each segment, a Principal Component Analysis (PCA) is carried out. If the eigenvalues indicate a high variance in two directions and a small variance in the remaining, the segment is assumed to be planar. In this case, the plane normal $\boldsymbol{n}$ corresponds with the eigenvector $\boldsymbol{v}_3$ belonging to the smallest eigenvalue λ_3. The parameter d missing for a complete plane equation can be calculated to

$$d = \boldsymbol{n}\boldsymbol{c} \tag{6}$$

using the centroid $\boldsymbol{c}$ of the segment.

In case the PCA does not yield a planar segment, we propose to use 3-D point descriptors in order to classify the segment. As result of our previous work in [23], Principal Curvatures (PC) are suited best for this purpose. PC is able to distinguish between planes, edges, corners, cylinders and spheres based on the minimum and maximum curvature. The reason for not processing planar segments with PC is that PCA is more efficient. As planes already have been identified and edges and corners do not represent surface regions, only cylinders and spheres are classified. However, as can be seen in the results section, spheres cannot be robustly distinguished from more complex structures in the environment. Thus, we introduce the complex class. Cylinders identified by PC can also be expressed by parameters (radius r, height h and pose). The pose is defined as a coordinate frame K with its z-axis

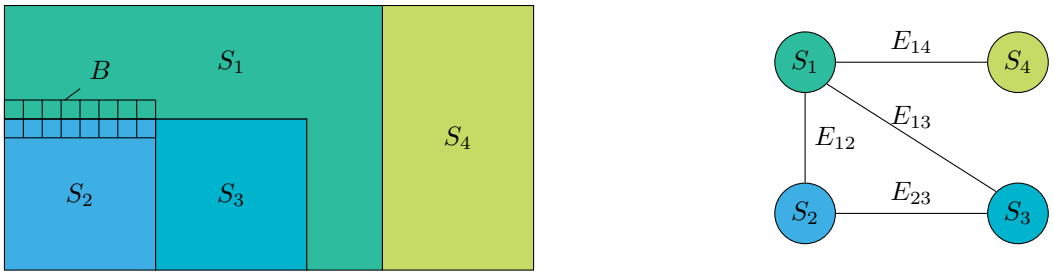


Fig. 3: The graph structure.

z_k being the symmetry axis of the cylinder and its x-axis x_k pointing towards the centroid $\boldsymbol{c}$ of the hull. Using the results of PC, the symmetry axis corresponds with the direction of the averages minimum curvature. This can be derived by a PCA over the points curvatures. The x-axis is found by determining a vector perpendicular to z_k, pointing to $\boldsymbol{c}$. In order to find r, all points of the segment are transformed to K. By only taking into account the x- and y-values, a circle can be fit (using RANSAC) to the points which has the same radius as the cylinder. Finally, h is separated into two values h_{min} and h_{max}. These can be derived from the segment points p_i using

$$h_{max} = \max \boldsymbol{p}_{i,z} \tag{7}$$

$$h_{min} = \min \boldsymbol{p}_{i,z}. \tag{8}$$

As most of the segments consist of concave polygons or incomplete cylinders, just deriving the shape parameters is not sufficient. Also, the contour of the hull has to be considered. Thus, the border points of each segment are ordered and re-sampled in order to get a hull with equally distributed points. This is done efficiently using pre-computed masks for line-following of the border points.

Fig. 4 shows the processing sequence on a table scene. The refinement can be observed on the curved surfaces of the bowls. After refinement, each surface consists of only one segment. The classified point cloud shows planes (light blue), cylinders (green) and complex structures (dark blue).

IV. Results

The method proposed (called GRG from now on) is evaluated on the publicly available IPA dataset[1], consisting of eight indoor scenes. The scenes show a table with objects upon, an office scene, a crowded cupboard and a kitchen. Two range set-ups (close and far) are chosen in order to evaluate distance related accuracy. The scenes were recorded with an RGB-D camera (Asus XTion Pro Live). In a first evaluation step, the segmentation and classification accuracy of the algorithms are tested using the ground truth of the scenes. Second, a comparative evaluation to state of the art (RANSAC based plane extraction) is shown. For all evaluation steps, the parameters from Table I are used.

For the segmentation accuracy, true and false segments are counted. A segment is false if it is over- or under-segmented.

[1]http://www.care-o-bot-research.org/contributing/data-sets

TABLE I: Parameters for evaluation.

Parameter	Value
Normal Estimation	
s_f	2
s_p	2
s_r	8
Segmentation	
c	, $0.006m^{-1}$
α_{th}	0.61rad
Refinement	
β_{th}	0.87rad
P_{th}	0.8

An under-segmented region, actually consisting of n surfaces is counted as $n-1$ false segments. Table II shows the accuracy values R for all scenes. An overall segmentation accuracy of 87.3% is achieved. The close-range scenes yield better results, as expected.

To evaluate the classification accuracy, the correctly and wrongly classified segments for each scene and each class are counted, e.g. true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). Table III shows the confusion matrices for all scenes. Most of the values are on the main diagonal which means that they are TP. Especially planes are classified with a high accuracy. In order to have the accuracy with comparable numbers, we calculate the micro and macro average values for precision, recall as well as the F-score according to

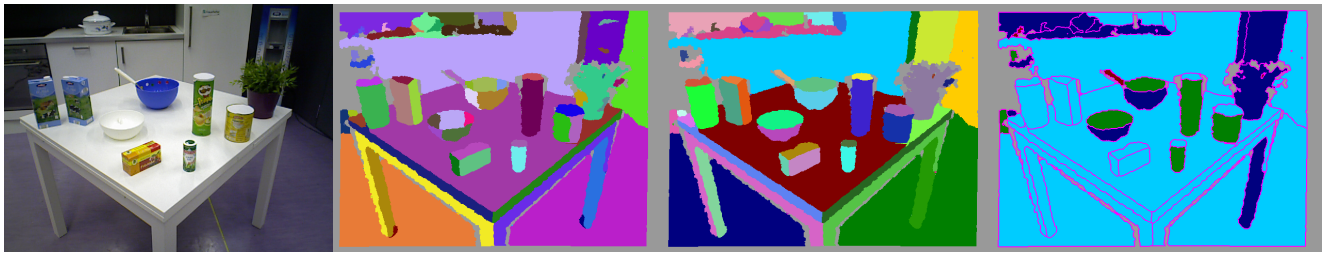
$$R_{\text{mic}} = P_{\text{mic}} = F_{\text{mic}} = \frac{\sum_i^k A_{ii}}{\sum_i^k \sum_j^k A_{ij}} = \frac{\sum_i^k TP_i}{\sum_i^k (TP_i + FN_i)} \tag{9}$$

$$R_{\text{mac}} = \frac{1}{k} \sum_i^k \frac{A_{ii}}{\sum_j^k A_{ij}} = \frac{1}{k} \sum_i^k \frac{TP_i}{TP_i + FN_i} \tag{10}$$

$$P_{\text{mac}} = \frac{1}{k} \sum_i^k \frac{A_{ii}}{\sum_j^k A_{ji}} = \frac{1}{k} \sum_i^k \frac{TP_i}{TP_i + FP_i} \tag{11}$$

$$F_{\text{mac}} = \left(1 + \gamma^2\right) \cdot \frac{P_{\text{mac}} \cdot R_{\text{mac}}}{(\gamma^2 \cdot P_{\text{mac}}) + R_{\text{mac}}} \tag{12}$$

with entries of the confusion matrix A_{ij}, k classes and $\gamma = 1$ [25]. Table IV depicts the values for all scenes. It can be seen that except for the kitchen and office far scenes (whose data have the highest noise) all values are above 0.7. Visual results are depicted in Fig. 5. Most of the planes are classified



(a) RGB image (b) Segmentation (c) Refinement (d) Surface classification

Fig. 4: Example for the processing steps.

TABLE II: Accuracy of the segmentation.

Scene	No. of segments	No. of correct segments	over-segmented	under-segmented	R (%)
kitchen close	23	21	1	1	91.3
kitchen far	14	10	1	3	71.4
table close	41	39	1	1	95.1
table far	28	23	4	1	82.1
office close	24	22	0	2	91.7
office far	35	30	1	4	85.7
cupboard close	38	36	0	2	94.7
cupboard far	51	44	1	6	86.3
average (σ)					87.3 ($\pm$ 0.079)

TABLE IV: Accuracy of the classification.

Scene	R_{mic}	R_{mac}	P_{mac}	F_{mac}
kitchen close	0.923	0.958	0.917	0.937
kitchen far	0.917	0.633	0.556	0.888
table close	1	1	1	1
table far	0.871	0.939	0.879	0.908
office close	0.958	0.982	0.833	0.902
office far	0.872	0.696	0.678	0.687
cupboard close	0.95	0.889	0.905	0.897
cupboard far	0.848	0.846	0.730	0.784
Average	0.917	0.868	0.812	0.875

correctly. In the *far* scenes, distinction between cylinders and complex structures is less clear because of sensor noise.

Timings were measured on an Intel Core i7-2600 CPU with 16 GB RAM, running on Ubuntu Linux 10.10, 64-Bit. Without special optimization such as parallelization or usage of GPU, processing of one point cloud with a resolution of 640x480 took 0.2 s on average. RANSAC proved to be comparable in speed. However, down-sampling of the point cloud using a voxel filter wit a resolution of 0.03 m was necessary. MPS is the fastest algorithm with a processing time of 0.028 s per frame. However, if we modify our algorithm to only output planar segments, the processing time is reduced to 0.048 s which is comparable to MPS. This shows, that our approach is capable of performing in real-time.

In a second evaluation step, our approach is compared to state of the art methods. The first one is using RANSAC based extraction of planes as presented in [26], the other one is the Multi Plane Segmentation (MPS) approach from the PCL library. As these methods are not able to extract further surface classes, only the planar segmentation and classification is evaluated on the same data set then before. We measure the error of the plane parameters as well as the contour errors of the two methods.

The results of the comparative evaluation are shown in Table V. Measurements show that the percentage of planes found is in general rather low for RANSAC. This is due to the inability to robustly extract small planes. On the contrary, GRG finds most of the planes in each scene, sometimes 100% of the planes are segmented correctly. MPS performs better than RANSAC but still only finds roughly 50% of the planes. The normal angle error α is evaluated using both the average and the median. The average error favors RANSAC and MPS. This is because some of the planes found by GRG have a huge deviation while the other methods only find larger planes which are more stable. The median, however, shows that the majority of the GRG segments have a very small error. The error of the area ΔA circumscribed by the polygon contours is used to show over- and under-segmentation. It can be seen that RANSAC produces a significant difference in the segment area. This is because of the non-selective behavior. MPS is also far worse than GRG because it tends to over-segment scenes.

In addition to the measurements, a visual comparison is given in Fig. 5. It shows the RGB image of the scene, the labeled ground truth and the results of MPS and GRG. RANSAC is omitted because it showed the weakest performance. MPS yields poor results in some scenes, especially in the far set. Significant over-segmentation can be observed in the *table close* scene. The pictures of GRG resemble the ground truth very well. Most of the planes, also small ones are extracted correctly.

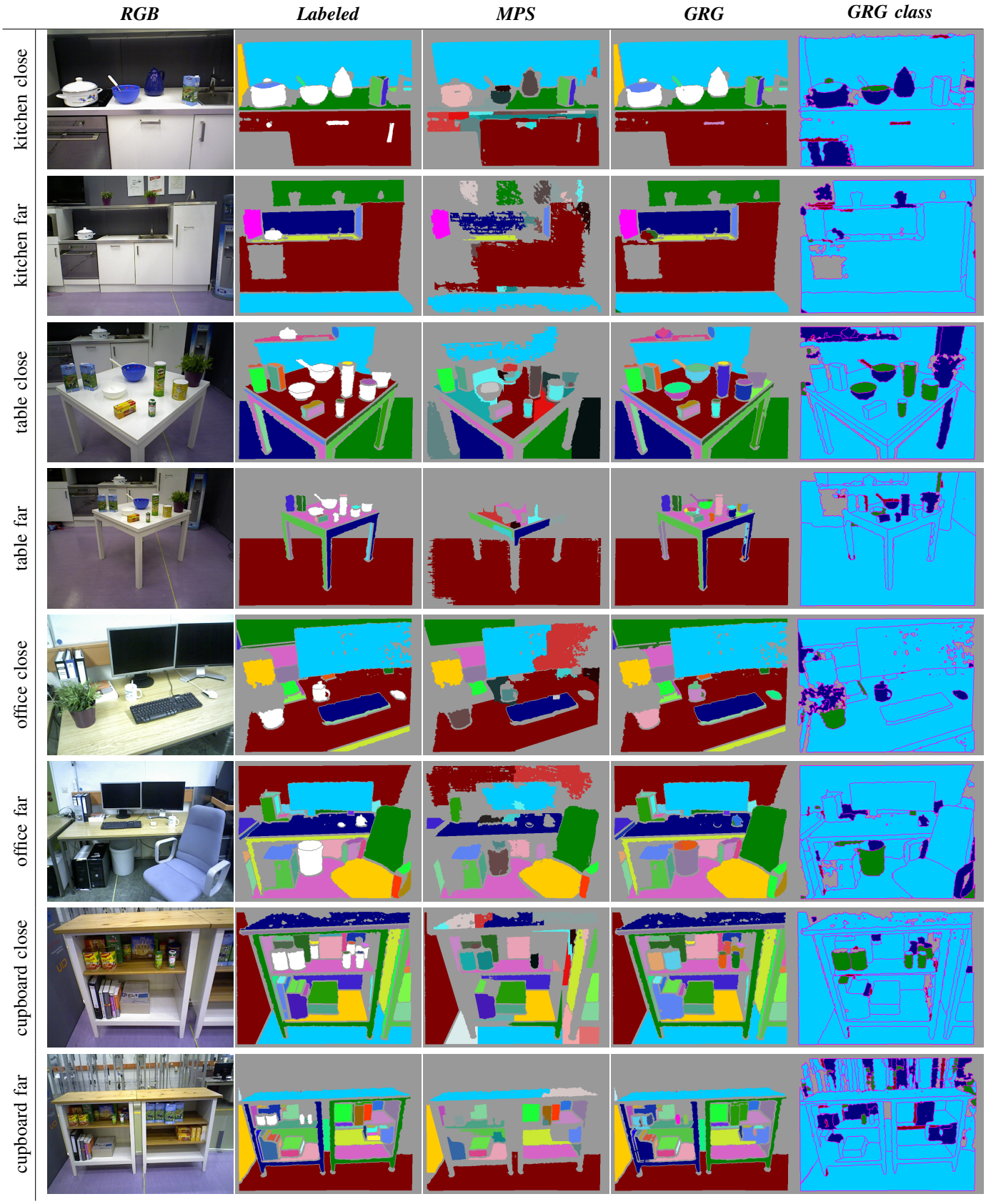


Fig. 5: Comparison of our approach to state of the art. Classification result of GRG.

TABLE III: Confusion matrices for the classification of planes (P), cylinders (Cy) and complex structures (Co). Rows depict estimated, columns predicted values.

(a) kitchen close

	P_p	Cy_p	Co_p
P_e	7		
Cy_e		2	
Co_e	1		3

(b) kitchen far

	P_p	Cy_p	Co_p
P_e	9		
Cy_e		0	
Co_e	1		2

(c) table close

	P_p	Cy_p	Co_p
P_e	23		
Cy_e		5	
Co_e			10

(d) table far

	P_p	Cy_p	Co_p
P_e	18		
Cy_e		2	
Co_e	4		7

(e) office close

	P_p	Cy_p	Co_p
P_e	18		
Cy_e	1	1	
Co_e			4

(f) office far

	P_p	Cy_p	Co_p
P_e	26		
Cy_e	1	1	1
Co_e		3	7

(g) cupboard close

	P_p	Cy_p	Co_p
P_e	29		
Cy_e		4	
Co_e		2	5

(h) cupboard far

	P_p	Cy_p	Co_p
P_e	30		5
Cy_e	1	1	1
Co_e			8

TABLE V: Comparative evaluation of RANSAC, MPS and our approach (GRG).

	Planes found (%)			Avg. $\Delta\alpha$ (rad)			Median $\Delta\alpha$ (rad)			$\Delta A(m^2)$		
Scene	**MPS**	**RANSAC**	**GRG**	**MPS**	**RANSAC**	**GRG**	**MPS**	**RANSAC**	**GRG**	**MPS**	**RANSAC**	**GRG**
kitchen close	45.5	27.2	100	0.014	0.038	0.017	0.041	0.047	0	0.077	0.138	0.003
kitchen far	100	100	100	0.034	0.050	0	0.045	0.042	0	0.290	0.234	0.019
table close	39.1	21.7	100	0.016	0.013	0.025	0.035	0.011	0	0.077	0.105	0.006
table far	28.6	21.4	92.9	0.090	0.026	0.259	0.139	0.03	0.01	0.139	0.123	0.015
office close	50.0	18.8	100	0.016	0.059	0.227	0.029	0.017	0	0.027	0.083	0.001
office far	53.3	26.7	96.7	0.046	0.139	0.058	0.091	0.063	0	0.058	0.103	0
cupboard close	50.0	10.5	92.1	0.015	0.130	0.067	0.027	0.047	0	0.020	0.073	0.001
cupboard far	43.5	8.7	87.0	0.026	0.140	0.046	0.054	0.151	0	0.015	0.037	0.005
Average	51.3	29.4	96.1	0.032	0.074	0.087	0.052	0.051	0	0.054	0.112	0.006

☐ worst fair best

V. Conclusions and Future Work

In this paper, we presented a novel method for segmentation and surface classification on ordered point clouds. The segmentation is done using region growing with point normals as similarity criterion. A refinement step utilizing a graph structure reduces over-segmentation. PC determines the underlying surface class in a per-point manner. It is able to distinguish between planes, cylinders and complex structures. Parameters of the geometric primitives are derived from the segment data. This leads to a compact data representation with enriched information. We showed in an elaborate evaluation on indoor scenes that our approach performs well both during segmentation and classification. Measurements of computation speed prove that our method is able to perform in real-time on full sensor resolution. In a comparison to state of the art, our approach revealed better accuracy and efficiency.

Future steps are to deal with the complex surface structures. Polynomial representations or meshes might be suitable for describing these. Furthermore, more sophisticated decision methods to derive the segment surface class from the point surfaces might further improve performance. Also, optimized implementation is a goal to be able to process point clouds at the full rate of 30Hz.

Acknowledgment

This research was partly funded from the EU FP7-ICT-287624 Acceptable robotiCs COMPanions for AgeiNg Years (ACCOMPANY).

References

[1] R. B. Rusu, N. Blodow, Z. Marton, A. Soos, and M. Beetz, "Towards 3d object maps for autonomous household robots," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 2007, pp. 3191–3198.

[2] K. Georgiev, R. T. Creed, and R. Lakaemper, "Fast plane extraction in 3d range data based on line segments." IEEE, 2011, pp. 3808–3815.

[3] M. Yang and E. Lee, "Segmentation of measured point data using a parametric quadric surface approximation," *Computer-Aided Design*, vol. 31, pp. 449–457, 1999.

[4] S. Suzuki, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, pp. 32–46, 1985.

[5] Y. Hsiao, C. Chuang, J. Jiang, and C. Chien, "A contour based image segmentation algorithm using morphological edge detection," in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, vol. 3, 2005, p. 29622967.

[6] A. Sappa and M. Devy, "Fast range image segmentation by an edge detection strategy," in *3dim*, 2001, p. 292.

[7] A. Bultheel and R. Cools, "Extraction of closed feature lines from point clouds based on graph theory," *Nieuw Archief voor Wiskunde*, vol. 5, pp. 24–28, 2008.

[8] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments," in *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, 2009, pp. 1–6.

[9] J. Biswas and M. Veloso, "Fast sampling plane filtering, polygon construction and merging from depth images," in *RSS, RGB-D Workshop*, 2011.

[10] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, pp. 915–926, 2008.

[11] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, pp. 214–226, 2007.

[12] C. J. Taylor and A. Cowley, "Fast scene analysis using image and range data," in *IEEE International Conference on Robotics and Automation*, 2011.

[13] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nchter, "The 3d hough transform for plane detection in point clouds: A review and a new accumulator design," *3D Research*, vol. 2, 2011.

[14] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using rgb-d cameras," in *Proceedings of the 15th RoboCup International Symposium*, 2011.

[15] T. Rabbani, F. van den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, pp. 248–253, 2006.

[16] Q. Zhan, Liang Yu, and Y. Liang, "A point cloud segmentation method based on vector estimation and color clustering," in *Information Science and Engineering (ICISE), 2010 2nd International Conference on*. IEEE, 2010, pp. 3463–3466.

[17] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," 2009.

[18] K. Koster and M. Spann, "Mir: an approach to robust clustering-application to range image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 430–444, 2000.

[19] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak, "Fast plane detection and polygonalization in noisy 3d range images," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 3378–3383.

[20] R. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1–4.

[21] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 1848–1853.

[22] Z. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz, "General 3d modelling of novel objects from a single view," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3700–3705.

[23] G. Arbeiter, S. Fuchs, R. Bormann, J. Fischer, and A. Verl, "Evaluation of 3d feature descriptors for classification of surface geometries in point clouds," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012.

[24] K. Khoshelham, "Accuracy analysis of kinect depth data," in *ISPRS Workshop Laser Scanning*, vol. 38, 2011, p. 1.

[25] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to information retrieval*. New York: Cambridge University Press, 2008.

[26] G. Arbeiter, R. Bormann, J. Fischer, M. Hägele, and A. Verl, "Towards geometric mapping for semi-autonomous mobile robots," in *Spatial Cognition VIII*, ser. Lecture Notes in Computer Science, C. Stachniss, K. Schill, and D. Uttal, Eds., vol. 7463. Springer, 2012, pp. 114–127.