3D Point Cloud Segmentation Oriented to The Analysis of Interactions

Xiao Lin
Image Processing Group
Technical University of Catalonia
(UPC)
Barcelona, Spain

Josep R.Casas
Image Processing Group
Technical University of Catalonia
(UPC)
Barcelona, Spain

Montse Pardás
Image Processing Group
Technical University of Catalonia
(UPC)
Barcelona, Spain

***Abstract*—Given the widespread availability of point cloud data from consumer depth sensors, 3D point cloud segmentation becomes a promising building block for high level applications such as scene understanding and interaction analysis. It benefits from the richer information contained in real world 3D data compared to 2D images. This also implies that the classical color segmentation challenges have shifted to RGBD data, and new challenges have also emerged as the depth information is usually noisy, sparse and unorganized. Meanwhile, the lack of 3D point cloud ground truth labeling also limits the development and comparison among methods in 3D point cloud segmentation. In this paper, we present two contributions: a novel graph based point cloud segmentation method for RGBD stream data with interacting objects and a new ground truth labeling for a previously published data set [1]. This data set focuses on interaction (merge and split between 'object' point clouds), which differentiates itself from the few existing labeled RGBD data sets which are more oriented to Simultaneous Localization And Mapping (SLAM) tasks. The proposed point cloud segmentation method is evaluated with the 3D point cloud ground truth labeling. Experiments show the promising result of our approach.**

I. Introduction

Segmentation is an essential task in computer vision. It usually serves as the foundation for solving higher level problems such as object recognition, interaction analysis and scene understanding. Traditionally, segmentation is defined as a process of grouping homogeneous pixels into multiple segments on a single image, which is also known as low level segmentation. The obtained segments are somehow more homogeneous and more perceptually meaningful than the raw pixels. Based on that, the concept of semantic segmentation/labeling is proposed. It is devoted to segment an image into regions which correspond to meaningful objects in the scene. To achieve this goal, high level knowledge is usually incorporated into the segmentation process. For instance, object models are used in semantic segmentation for constrained scenes like human body detection and pose recognition [2], temporal information is employed when stream data is available [3] and label contextual information is exploited in other approaches [4], [5]. However, segmentation methods based on the 2D image are limited, as a lot of valuable information about the shape and geometric layout of objects is lost when a 2D image is formed from the corresponding 3D world. The emergence of cheap consumer depth sensors, like Kinect, makes it easier to access depth information. This offers the potential to segment objects considering the richer geometric information in actual 3D. Similar to 2D segmentation methods, there are also several 3D approaches working at low level segmentation, such as super voxels[6], or region growing [7]. They focus on grouping 3D points represented by local features into meaningful 3D segments, while better preserving object boundaries than similar 2D methods due to significative boundary information provided by depth maps. Approaches based on geometric shapes like planes [8], [9] and cylinders [10] are also proposed to achieve high level segmentation for objects with certain shapes. Graph based methods, such as Markov Random Fields (MRF) and Conditional Random Fields (CRF) are employed to solve 3D semantic segmentation because of their convenient property of merging low-level context with object level class knowledge [11]. We notice that most 3D segmentation approaches are inspired or extended from 2D methods while benefiting from the richer information in 3D data. However, 3D data also brings new challenges, such as point clouds are usually noisy, sparse and unorganized. Furthermore, the lack of ground truth labeling for 3D point clouds makes it hard to evaluate the segmentation results and also impairs the machine learning based approaches. The few existing data sets with ground truth labeling mainly focus on static indoor scenes. For instance, Cornell RGBD data set [12] contains 24 labeled office scenes and 29 labeled home scenes. Similarly, NYU data set [13] contains 464 different indoor scenes with 1449 densely labeled frames. TUM data set [14] provides more indoor scene RGBD data but only the ground truth of camera pose is provided.

These data sets are more suitable for SLAM tasks and they share the same data capture mode which consists in capturing a static scene with a moving camera. However, a large proportion of our visual experience involves analyzing the interactions between objects. Semantic segmentation in scenes should have the ability to cope with interacting objects, which might be deformable, and, if possible, without previously learned object models. In this context we present a temporal consistency guided graph based 3D point cloud segmentation approach for point cloud stream data, which works on a low level graph representation of the point cloud. Then we analyse

the impact from different low level graph building methods on the proposed approach. We present, as a second contribution, the ground truth labeling of the 3D foreground point cloud for a previously published human manipulation data set [1]. The experiments are performed on this newly labeled data set.

II. Graph based 3D point cloud segmentation

Given the color and depth data captured by a consumer depth sensor, we can transform the per pixel distances provided in the depth image into a 3D point cloud $C_I \subseteq R^3$ with the camera parameters. As we are more concerned about the foreground cloud $C_{fg} \subseteq R^3$, we focus on the interest area in 3D space. Points in the interest area are then treated as the foreground point cloud. Taking the foreground point cloud at frame t as the input data, a graph representation is constructed from the input data $f(C_{fg}) \rightarrow G(v,e)$ via a graph building method f, where v are the vertices or nodes and e the edges of the graph. The graph representation of input data simplifies the segmentation task. Its aim is to group homogeneous points on the cloud into nodes, while also preserving the available boundary information. In this manner, segmentation decisions are made at a higher level instead of at point level, which not only reduces the problem scale but also makes segmentation more robust to noise. The segmentation problem is then converted to a graph node labeling task in order to minimize the energy of assigning a label to a node considering both spatial and feature homogeneity. In the case of video segmentation, the temporal consistency can also be taken into account. Therefore, in our approach, the energy function is defined as:

$$E(L) = E_{data} + E_{smooth} \tag{1}$$

In which the total energy E for a labeling L is specified as the summation of two energy terms, data cost E_{data} and smoothness cost E_{smooth}. The data term is defined as the summation of costs of assigning a label l_{v_i} to a node v_i.

$$E_{data} = \sum_{v_i \in G} D_{v_i}(v_i, l_{v_i}) \tag{2}$$

The smoothness term is defined as the summation of costs of assigning labels l_{v_i} and l_{v_j} respectively to nodes v_i and v_j which are connected by an edge e_i. This energy enforces the smoothness between neighboring nodes on the graph.

$$E_{smooth} = \sum_{e_i \in G} S_{e_i}\left(l_{v_i}, l_{v_j}\right) \tag{3}$$

To exploit spatial smoothness and temporal consistency, D_{v_i} and S_{e_i} are defined on both the previous information and the current data. Given the graph $G'(v', e')$ in the previous frame and corresponding labels $L' \ni \left(v_i', l_{v_i}'\right)$, the two energy terms for labeling the current graph $G(v,e)$ with L are formulated as:

$$D_{v_i}(v_i, l_{v_i}) = \min_{\left(v_i', l_{v_i}\right) \in L'} dist\left(F(v_i), F(v_i')\right) \tag{4}$$

$$S_{e_i}\left(l_{v_i}, l_{v_j}\right) = E_{cur}\left(l_{v_i}, l_{v_j}\right) + E_{prev}\left(l_{v_i}, l_{v_j}\right) \tag{5}$$

The data cost for assigning label l_{v_i} to node v_i is represented as the minimum distance in the feature space between node v_i and $v_i' \in G'$, which was labeled with l_{v_i}. $F(\cdot)$ stands for the feature extraction operation which yields a feature vector for the input node by concatenating 3 components, color feature $H_c(v_i)$ (color histogram in LAB color space), shape feature $H_s(v_i)$ (Histogram of Oriented Normal Vector–HODV [15]) and 3D position $v_i(x_{v_i}, y_{v_i}, z_{v_i})$ (3D coordinates of the node centroid). $H_c(v_i)$ and $H_s(v_i)$ are normalized to the range $[0,1]$. The distance for each component is calculated separately.

$$\begin{aligned} dist\left(F(v_i), F(v_i')\right) &= \omega_c \cdot dist_i\left(H_c(v_i), H_c(v_i')\right) \\ &+ \omega_s \cdot dist_i\left(H_s(v_i), H_s(v_i')\right) + \omega_p \cdot dist_e\left(v_i, v_i'\right) \end{aligned} \tag{6}$$

We employed histogram intersection distance [16] $dist_i(\cdot)$ for color and shape features, and Euclidean distance $dist_e(\cdot)$ for centroid position of two nodes. Note that the Euclidean distance between node centroids is normalized to $[0,1]$ with a sigmoid function. The histogram intersection distance is formulated in (7), where H_j stands for the jth dimension of the Histogram H.

$$dist(H, H') = \sum_j \min\left(H_j, H_j'\right) \tag{7}$$

Then $dist(\cdot)$ is defined as the weighted sum of the three distances in (6), where ω_c, ω_s, ω_p are the weighting factors for the three components. In our experiments, they are all set to 1/3 empirically.

The smoothness term, defined in (5) is described as the summation of the energy in the current frame and the previous frame. $E_{cur}\left(l_{v_i}, l_{v_j}\right)$ describes the difference between two connected nodes v_i and v_j on the current graph G in the feature space.

$$E_{cur}\left(l_{v_i}, l_{v_j}\right) = dist\left(F(v_i), F(v_j)\right) \tag{8}$$

whereas $E_{prev}\left(l_{v_i}, l_{v_j}\right)$ is the difference between edge e_i connecting v_i and v_j and edge e_i' connecting their most similar nodes v_i^* and v_j^* with the same label l_{v_i} and l_{v_j} on the graph from the previous frame. The difference is computed by comparing the position between e_i and e_i' (the summation of the distances between corresponding end-points), and the length of the two edges. The distance between two edges is normalized to $[0,1]$ with a sigmoid function. The difference in the length of edges is defined as a normal distribution $\mathbb{N}\left(dist_e(v_i, v_j) \mid \mu, \sigma^2\right)$ in which μ is the length of e_i' which equals the Euclidean distance between v_i^* and v_j^*, $dist_e\left(v_i^*, v_j^*\right)$, σ is the predefined variation parameter and δ is a normalization factor. α is a weighting factor for balancing them. In our experiments, σ and α are set to $10cm$ and 0.5 respectively.

$$\begin{aligned} E_{prev}\left(l_{v_i}, l_{v_j}\right) &= \alpha * \left(1 - \mathbb{N}\left(dist_e(v_i, v_j) \mid \mu, \sigma^2\right)/\delta\right) \\ &+ (1-\alpha) * sigmoid\left(dist_e(v_i, v_i^*) + dist_e\left(v_j, v_j^*\right)\right) \end{aligned} \tag{9}$$

$$v_i^* = \arg \min_{\left(v_i', l_{v_i}\right) \in L'} dist\left(F(v_i), F(v_i')\right) \tag{10}$$

The energy defined above is then minimized via the graph cut method introduced in [17], [18], [19].

III. Graph building methods

As mentioned in Section II, the input point cloud is represented by a graph G, and the method used to build the graph is critical for the segmentation process in the next step. In the state of the art, several approaches to build graphs from unorganized point clouds have been used in different applications such as end-effector estimation [7], pose reconstruction [20], and video segmentation [6]. To distinguish which graph building method better supports the proposed segmentation strategy, we select two methods [7], [6], which are conceptually suitable for the proposed segmentation approach, and test them in the experiments.

The method in [7] employs level sets in RGB-D data to exploit connectivity over the depth surface. More specifically, it expands and includes a set of new points on the point cloud with respect to the previous level set, under the constraints of proximity, density and color. In this manner, the constructed graph represents a point cloud while preserving its topology and boundary information. In [6], the method works on the voxel representation of the input point cloud, which is parameterized by the voxel resolution R_{voxel}. Then a set of seeds are generated uniformly in the space with a seed radius $R_{seed} \gg R_{voxel}$. Taking the seeds as initialized cluster centers, a local k-means clustering is performed in a 39 dimension feature space which involves spatial coordinates, color and a local shape feature. It produces patches that adhere well to the object boundaries as it strictly enforces spatial connectivity in segmented regions.

IV. 3D point cloud ground truth labeling

We have generated the ground truth labeling for the 3D point clouds of a previously published human manipulation data set [1]. The original data set provides calibrated RGB-D video recorded using a Kinect device with 30 Hz frame rate and a resolution of $640 * 480$, and the foreground mask manually labeled on the color image. This provides the possibility to generate the 3D foreground point cloud labeling by back-projecting labeled pixels on the color image to 3D space. However, due to the noise on the depth image, points on the object boundary might not have the correct depth information, which results in an incorrect point cloud labeling. Fig.1(b) shows an example of the incorrect labeling when we back-project the pixel labeling to the 3D cloud (different colors show different foreground objects, red points correspond to unlabeled foreground points). In this case, the data set provides a foreground mask including a juice box, a cup and a corn flake box. To generate the correct 3D ground truth labeling, we need to face two problems:

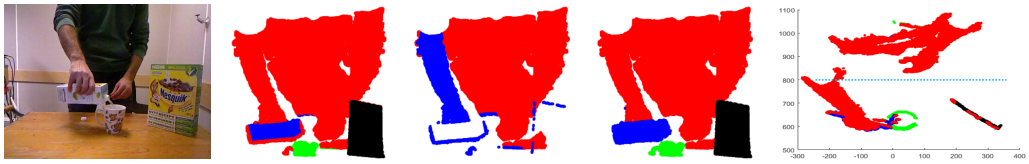
- The labeling obtained from the pixel labels on the color image is not complete (see the red points on the boundaries of the foreground objects) and the labeling from the pixel labels is not fully reliable (part of the label blue labeled-juice box is wrongly taking points on the human body).
- The pixels of the human body are not included in the foreground mask.

To address this problem, we need to add more information in order to obtain a better labeling. For this data set, we believe that the points with a distance to the camera higher than a threshold (800mm in the case of Fig.1(e)) belong to human body for sure. On the contrary, points within the foreground masks and with the distance lower than the threshold are well labeled. The rest of the other points which neither belong to human body nor to the objects, will be treated as unlabeled points. As shown in Fig.1(c), the red points are assigned to the human body and the blue points are categorized to be unlabeled. Then the problem is converted to a problem of assigning the unlabeled points to the few classes of labeled points, as now the labeled points can be trusted and the human body is partially labeled. We find that the back-projected pixel labeling for the majority of the frames have a clear boundary labeling when two objects attach to each other. As human body is the only manipulator in this scene, it has the highest frequency of attaching to other objects. Hence we employ the method proposed in [7] to expand the labeling from the original labeled human body point cloud in order to include the unlabeled points which are spatially connected to the origin. After that, the rest of the unlabeled points are assigned to objects based on distance. Fig.1(d) shows one example of the corrected 3D point cloud labeling. Note that manual correction is performed when the "elaborated" labeling correction strategy explained above fails (The number of manually labeled frames are lower than 10%, 36 frames out of 404 frames, for the proposed dataset). All the labeled 3D point cloud data and its ground truth data is publicly available on the website https://imatge.upc.edu/web/resources/human-manipulation-dataset.

V. Experiments

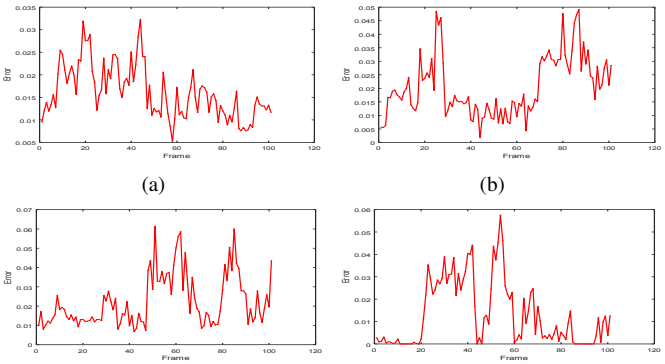
A. Segmentation result evaluation

To evaluate our 3D point cloud segmentation approach, we select 4 sequences with 3D point cloud ground truth labeling in the human manipulation data set. Each of them contains 101 frames. These 4 sequences vary from single attachment to multi-attachment, low motion to higher motion, double attached objects to multiple attached objects. We choose the super voxel based method [6] as our graph building method in this experiment. The evaluation metrics is 3D segmentation accuracy (3D ACCU) proposed in [21], which computes the fraction of a ground truth segment that is correctly classified in the approach. As the super voxel based graph building method works with the downsampled point cloud while the ground truth is labeled in the original cloud, we find K nearest neighbors for a point on the down sampled point cloud from the ground truth labeling and use a majority vote for the labels in the K nearest neighbors as the ground truth labeling for this point. In this experiment, we initialize the system with the



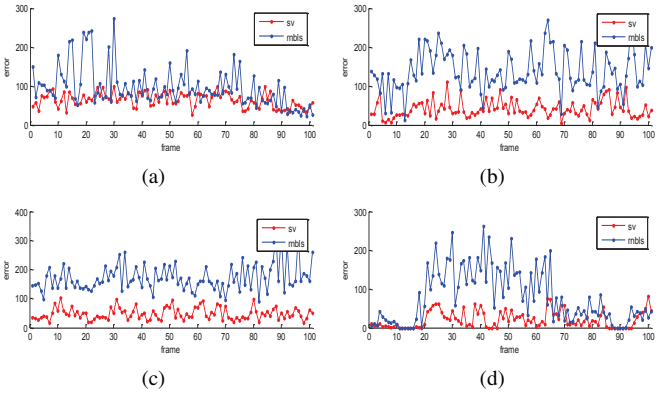
(a) (b) (c) (d) (e)

Fig. 1. (a) Color image (b) 3D ground truth labeling obtained from pixel labeling (c) Reliable points for human body and unlabeled points (d) Corrected 3D ground truth labeling (e) Top view of the point cloud in Fig.1(b). The blue dash line represents the distance threshold used in our strategy



(a) (b) (c) (d)

Fig. 2. Segmentation result for SV based graph building method in 4 different sequences, shown as error (vertical axis) per frame (horizontal)



(a) (b) (c) (d)

Fig. 3. Quantitative results in error per frame for different sequences (a)-(d). Red point/line represents SV, blue point/line represents RNBLS

ground truth as the previous information for the first frame. Fig. 2 shows the segmentation results of these 4 sequences. In each sub-figure, the percentage of segmentation error is plotted against the frame number. Our 3D point cloud segmentation approach achieves an overall 2% mean segmentation error while keeping the highest single frame segmentation error lower than 7%. Besides, the experiment result also shows the robustness of our approach regarding the noise in the previous information, as a steep decrease in segmentation error could be found in each of these sequences.

B. Comparison experiments on two graph building methods

These 4 sequences are also exploited in the comparison experiments to investigate the impact from different graph building methods. Two different graph building methods, restricted narrow band level set (RNBLS[7]) and super voxel (SV[6]) are employed to build the graph from the input foreground point cloud. Then we apply the same segmentation method on the constructed graphs. As introduced in Section II, the segmentation result of the previous frame is used to guide the current segmentation task in both data cost E_{data} and smoothness cost E_{smooth}. This means that the segmentation error in the previous frame will be propagated to the current frame. Therefore, in order to avoid error propagation along time and to compare the single frame performance between the two graph building methods in this experiment, we take the ground truth labeling of the previous frame as the previous frame segmentation to guide the current segmentation in both graph building methods. Since the number of points on the point cloud in each frame is the same for both graph building methods, we directly count the number of incorrectly segmented points compared to the ground truth labeling instead of the 3D ACCU. This makes the comparison results visually clearer, because the fraction of segmentation error usually turns to be a small number. Fig.3 shows the quantitative results for 4 different sequences representing the number of incorrectly segmented points per frame. The segmentation error for the super voxel based graph building method is plotted in red, and the result of RNBLS, in blue. Fig.4 shows four examples of qualitative segmentation results based on the two graph building methods. The first column in this figure is the original color images, the second column shows the results of SV and the third column are the result of RNBLS. Different objects in the scene are labeled in different color. The comparison experiments show that the super voxel based graph building method performs both qualitatively and quantitatively better than the RNBLS based one.

VI. Conclusion

In this paper, we have introduced a temporal consistency guided graph based 3D point cloud segmentation approach, which works on a low level graph representation of the point cloud. Besides, we contribute a new 3D point cloud ground truth labeling for the human manipulation data set introduced in [1]. This data set is more convenient for the analysis of object interactions than for SLAM tasks. To the best of our

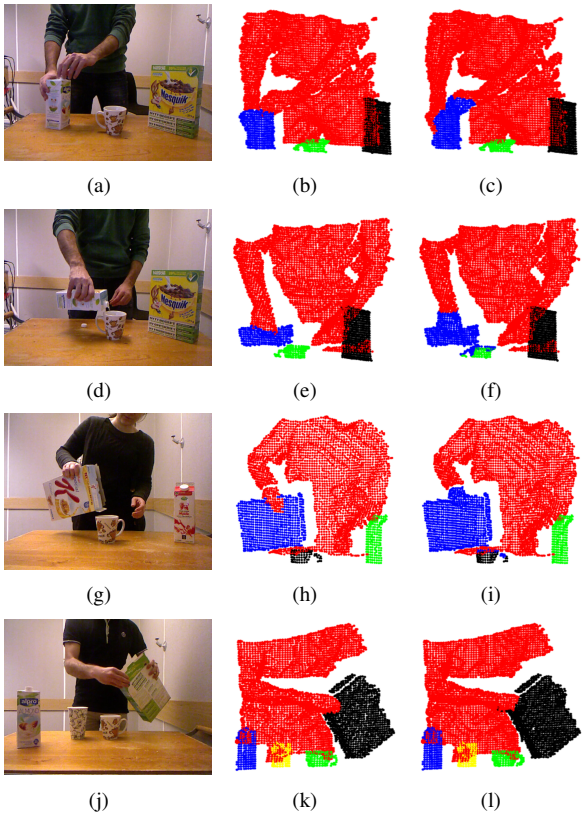


Fig. 4. Four segmentation result examples. (a)(d)(g)(j) show the color images, (b)(e)(h)(k) are the results of SV and (c)(f)(i)(l) are the results of RNBLS

knowledge, it is the first object interaction RGBD data set with 3D point cloud ground truth labeling.

The evaluation of the presented segmentation approach is done both in performance and relative to the impact of different graph building methods in the newly labeled data set. The proposed approach provides an overall 2% mean segmentation error while keeping the highest single frame segmentation error lower than 7%. Experiment results shows the robustness of the proposed method to signal noise and estimation errors in previous frames, and also proves the improved performance of super voxel strategies over RNBLS for experiments oriented to interaction analysis.

Acknowledgement: This work has been developed in the framework of the project TEC2013-43935-R, financed by the Spanish Ministerio de Economa y Competitividad and the European Regional Development Fund (ERDF). This work has received funding from Eusipco'11 organization

REFERENCES

[1] A. Pieropan, G. Salvi, K. Pauwels, and H. Kjellstrom, "Audio-visual classification and detection of human manipulation actions," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 3045–3052.

[2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, 2010.

[3] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2141–2148.

[4] X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán, "Multiscale conditional random fields for image labeling," in *Computer vision and pattern recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE computer society conference on*, vol. 2. IEEE, 2004, pp. II–695.

[5] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 1–15.

[6] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter, "Voxel cloud connectivity segmentation-supervoxels for point clouds," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2027–2034.

[7] X. Suau, J. Ruiz-Hidalgo, and J. R. Casas, "Detecting end-effectors on 2.5 d data using geometric deformable models: Application to human pose estimation," *Computer Vision and Image Understanding*, vol. 117, no. 3, pp. 281–288, 2013.

[8] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using rgb-d cameras," in *RoboCup 2011: robot soccer world cup XV*. Springer, 2011, pp. 306–317.

[9] B. Enjarini and A. Gräser, "Planar segmentation from depth images using gradient of depth feature," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4668–4674.

[10] Z.-C. Marton, L. Goron, R. B. Rusu, and M. Beetz, "Reconstruction and verification of 3d object models for grasping," in *Robotics Research*. Springer, 2011, pp. 315–328.

[11] R. B. Rusu, A. Holzbach, N. Blodow, and M. Beetz, "Fast geometric point labeling using conditional random fields," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 7–12.

[12] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," in *Advances in neural information processing systems*, 2011, pp. 244–252.

[13] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 746–760.

[14] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 573–580.

[15] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao, "Histogram of oriented normal vectors for object recognition with a depth sensor," in *Computer Vision–ACCV 2012*. Springer, 2012, pp. 525–538.

[16] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[17] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 11, pp. 1222–1239, 2001.

[18] V. Kolmogorov and R. Zabin, "What energy functions can be minimized via graph cuts?" *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 2, pp. 147–159, 2004.

[19] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 9, pp. 1124–1137, 2004.

[20] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt, "A data-driven approach for real-time full body pose reconstruction from a depth camera," in *Consumer Depth Cameras for Computer Vision*. Springer, 2013, pp. 71–98.

[21] C. Xu and J. J. Corso, "Evaluation of super-voxel methods for early video processing," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1202–1209.