Localizing and Segmenting Objects with 3D Objectness

Aitor Aldoma, Markus Vincze
Vision 4 Robotics, Technical University of Vienna
aldoma@acin.tuwien.ac.at

Federico Tombari
DEIS, University of Bologna

Walter Kropatsch
PRIP, Technical University of Vienna

Paper ID 22

Abstract. *This paper presents a novel method to localize and segment objects on close-range table-top scenarios sensed with a depth sensor. The method is based on a novel* objectness *measure that evaluates how likely a 3D region in space (defined by an oriented bounding box) could contain an object. Within a parametrized volume of interest placed above the table plane, a set of 3D bounding boxes is generated that exhaustively covers the parameter space. Efficiently evaluating — thanks to integral volumes and parallel computing — the 3D objectness at each sampled bounding box allows efficiently defining a set of regions in space with high probability of containing an object. Bounding boxes characterized by high objectness are then processed by means of a global optimization stage aimed at discarding inconsistent object hypotheses with respect to the scene. We evaluate the effectiveness of the method for the task of scene segmentation.*

1. Introduction and related work

Accurate robotic perception is a fundamental feature for most envisioned application scenarios related to service and industrial robotics. The capability of segmenting a scene perceived by a sensor onboard a robotic agent into a set of coherent patterns (or objects) is a classical - though challenging - step standing at the grounds of numerous tasks related to robotic perception such as 3D object recognition, point cloud registration, object grasping and manipulation. As commonly deployed onboard most robotic architectures, we assume the presence of a 3D perception system, acquiring RGB-D data (a color frame plus an associated *organized* point cloud), as well as that of a dominant plane in the scene, which can be represented by either the ground floor or a table on which objects are lying. The assumption of a dominant plane has been extensively used in the field of robotic perception to speed up segmentation such as in [6, 1, 2]. Other 3D segmentation methods without the dominant plane assumption are those presented in [8, 9]; even though they are more general than those constrained by the dominant plane assumption, they are characterized by a higher computational complexity.

Under these conditions, we have devised a novel algorithm aimed at automated localization of *salient* volumes from the data related to the scene currently in front of the robot. Our definition of saliency is driven by the concept of *objectness*, i.e. a portion of volume of the analysed 3D space is salient if the characteristics of the surface therein enclosed have a high probability of representing an object, and viceversa. To this aim, the first contribution of this work is the definition of an objectness measure for 3D data which can be computed on a 3D bounding box of generic dimensions, inspired from the work of Alexe et al. [4] that proposed an analogous measure for images. Based on our definition of objectness, we then propose an effective optimization framework to simultaneously detect the presence of several salient bounding boxes in a 3D scene, which is able to discard unrealistic object hypotheses such as objects intersecting one another or bounding boxes that do not fit tightly the object surface. Although not ex-

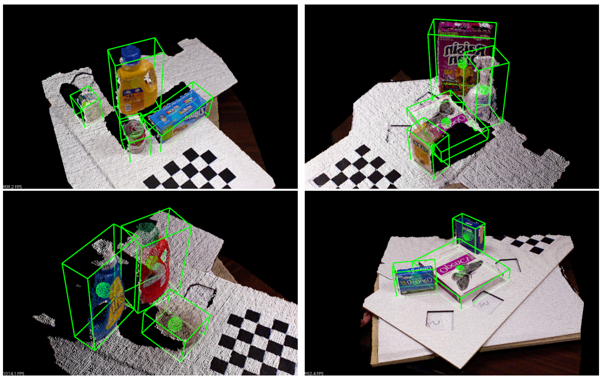


Figure 1: Typical results obtained by the proposed method. The green bounding boxes show the regions selected by the method that are likely to contain an object. The green sphere depict the center of the bounding box.

plicitly aimed at segmentation, in order to reproduce some quantitative and qualitative experimental analysis, we compare the results of our approach with those of state-of-the-art segmentation methods for point clouds, demonstrating the effectiveness of our proposal. We expect our method to prove useful as a pre-processing step of 3D object recognition algorithms, in order to reduce the number of false positives and improve the efficiency of current proposals relying on matching global - as well as local - 3D descriptors [10, 1, 2, 3, 11].

1.1. Objectness for images

Alexe et al. presented in [4] an *objectness* measure for color images in an attempt to evaluate the presence of an object without any specific object class knowledge. They present several image cues aimed at capturing the closed boundaries of objects, saliency as well as color contrast that are finally combined into a single objectness measure, to be computed on a 2D rectangular bounding box. The objectness measure is evaluated at different randomly sampled locations and proved to be useful in speeding up specific object class detectors.

In our case, the availability of 3D information provides powerful cues to reason about objects directly on the same 3D domain where the actual objects reside. This allows computing tight volumes enclosing the objects as well as the ability to reason about free and occupied space. Additionally, normals computed on the surface of the objects provide a powerful cue to assess surface continuity and smoothness. Throughout the work, we will show how this additional tools prove useful to evaluate the presence of an object within a closed region in 3D space.

2. Notation and preliminaries

For a scene of interest, let $\mathcal{M}$ represent the depth map acquired from the RGB-D sensor and $\mathcal{S}$, in sensor coordinates, the 3D point cloud, as seen from a certain viewpoint $\vec{vp}$, reconstructed from $\mathcal{M}$. We assume that $\mathcal{S}$ contains a dominant planar surface $\mathcal{P} = \{\vec{n}, d\}$, $\vec{n}$ being the normal to the plane, d being the distance to the global reference frame, on which objects lie upon. Using $\mathcal{P}$, we apply a rotation and translation to the global reference frame of $\mathcal{S}$ so that its z-axis is aligned with $\vec{n}$ and its origin is on the plane ($d = 0$). We are now able to compute the *Volume of Interest* (VoI), a region in the 3D space containing all objects of interest, by checking the maximum extensions of the points above the table. The VoI defines as well the region where the bounding boxes will be sampled and the objectness measure evaluated.

2.1. Complexity of the parameter space

Differently to [4], in a 3D domain each bounding box b is characterized by 9 degrees of freedom: $b = b(x, y, z, s_x, s_y, s_z, r_x, r_y, r_z)$, where (x, y, z) represents the reference corner of a bounding box, (s_x, s_y, s_z) its extension along the positive direction of the 3 axes and (r_x, r_y, r_z) its orientation. To reduce the complexity of the parameter space, we model only r_z (rotations about $\vec{n}$), assuming $r_x = r_y = 0$, this being motivated by the fact that most objects lying on a table are well contained by a bounding box with one plane parallel to the dominant planar surface $\mathcal{P}$. Additionally, since objects lie on $\mathcal{P}$, it is possible to set $z = 0$, this resulting in 6 dimensions to be sampled and yielding $b = b(x, y, s_x, s_y, s_z, r_z)$ (the dependency of b from its independent variables will be dropped hereinafter for conciseness of notation).

Even after reducing the complexity of the problem, the number of bounding boxes that need to be evaluated remains high. However, thanks to *Integral Volumes* (IV) [5] it is possible to evaluate in constant time sums of elements (points, edges, etc.) contained in the volume of space within (x, y, z) and $(x + s_x, y + s_y, z + s_z)$. To model r_z, it is possible to rotate $\mathcal{S}$ at different angles ($0° \leq r_z < 90°$) and compute additional IVs for each ρ_i. Modeling r_z allows to obtain bounding boxes that enclose the object tightly. Next section discusses the different IVs that need to be computed to represent the cues required for the 3D objectness measure. Where not differently stated, the IVs are computed at a resolution of 1 cm.

2.2. Sampling bounding boxes

To cover the VoI, we perform an exhaustive sampling of the parameter space generating a bounding box b at each possible location. Corners defined by triplets (x,y,$z = 0$) are sampled every 2 cm along x and y directions; the bounding box extension defined by triplets (s_x, s_y, s_z) are sampled respectively every (2,2,1) cm; finally, r_z - the rotation angle about z - is sampled every $5°$. We include a prior on the minimum and maximum size of objects, thus restricting s_x, s_y, s_z to be within the range $[3; 45]$cm. Such parameterization typically results in about 600 million bounding boxes to be evaluated for each scene acquired by the sensor. Thanks to the IVs and the parallel computing capabilities of GPU devices, it is possible to evaluate such amount of bounding boxes efficiently.

2.3. Occlusion and occupancy volumes

Previous to the definition of the 3D objectness measure, we need to introduce the concepts of occlusion volume and occupancy volumes. These are binary volumetric representations with an extension equal to that of the VoI and will allow us, later on, to derive important cues such as the free space inside a bounding box.

The occupancy volume – $\mathcal{V}$, is simply a binary representation of $\mathcal{S}_p$ where a voxel takes the value of 1 when at least a point $p \in \mathcal{S}_p$ falls inside the voxel boundaries, 0 otherwise. The occlusion volume – $\mathcal{V}_{\mathcal{O}}$, is likewise a binary set of voxels encoding whether a voxel is visible from the viewpoint $\vec{vp}$ or not, respectively taking values 0 and 1. To build $\mathcal{V}_{\mathcal{O}}$ we make use of the depth map $\mathcal{M}$ and the VoI previously computed. Concretely, we build a dense point cloud $\mathcal{C}_{\mathcal{O}}$ spanning the VoI with a resolution of 1cm. Afterwards, we backproject each point $p_i \in \mathcal{C}_{\mathcal{O}}$ to $\mathcal{M}$ using the calibration parameters of the sensor and reject all p_i with a depth value lower than the corresponding depth value in $\mathcal{M}$. The last step removes all visible points in $\mathcal{C}_{\mathcal{O}}$ and allows us to generate $\mathcal{V}_{\mathcal{O}}$ by simply checking if the voxel is empty or not. The middle part of Figure 2 shows an occlusion volume for a specific scene.

3. 3D Objectness

This section presents the cues as well as how they are combined together to define the 3D objectness measure. Similar to [4], the goal behind such cues is to capture the closed boundaries of objects in order to obtain high values for bounding boxes that contain an object entirely and that enclose it tightly.

3.1. Edge density

The first cue under consideration regards edges. Differently from [4], by reasoning in the 3D space it is possible to define a much richer set of edges than on the image plane. Specifically, we have deployed an edge extraction algorithm, available on the Point Cloud Library (PCL) [1], which is able to extract and discriminate between edges derived by surface curvature variations (blue), edges causing occlusions (orange), edges caused by occlusions (green) as well as scene border edges (red); reported colors refer to the left image of Figure 2, where a sample scene is depicted together with the extracted edges. The whole set of edges associated to a bounding box b will be referred to as $\varepsilon(b)$.

A nice property of such edges is that they are usually found on the surface of the objects. Therefore, when a bounding box encloses a high number of edges compared to the area of its visible faces, this intuitively represents a strong cue for the presence of an object inside it. Observe that from a certain viewpoint, there will be always at most 3 faces of a bounding box that are visible. We thus define the edge density cue δ_i for a bounding box b as follows:

$$\delta_i(b) = \frac{|\varepsilon(b)|}{a(b)} \tag{1}$$

where $| \cdot |$ is the cardinality operator and $a(b)$ is the visible area of b.

3.2. Outer edges

A second cue derived as well from edges is aimed at penalizing bounding boxes that contain edges in their immediate surroundings. The neighborhood of a bounding box b is defined by a bigger bounding box $b_\epsilon = b(x, y, s_x + s_{x,\epsilon}, s_y + s_{y,\epsilon}, s_z + s_{z,\epsilon}, rz)$ (we set $s_{x,\epsilon} = s_{y,\epsilon} = 2$cm and $s_{z,\epsilon} = 4$cm). This cue is represented by the ratio between the number of edges inside b, and the number of edges inside the

[1] www.pointclouds.org/blog/gsoc12/cchoi/index.php

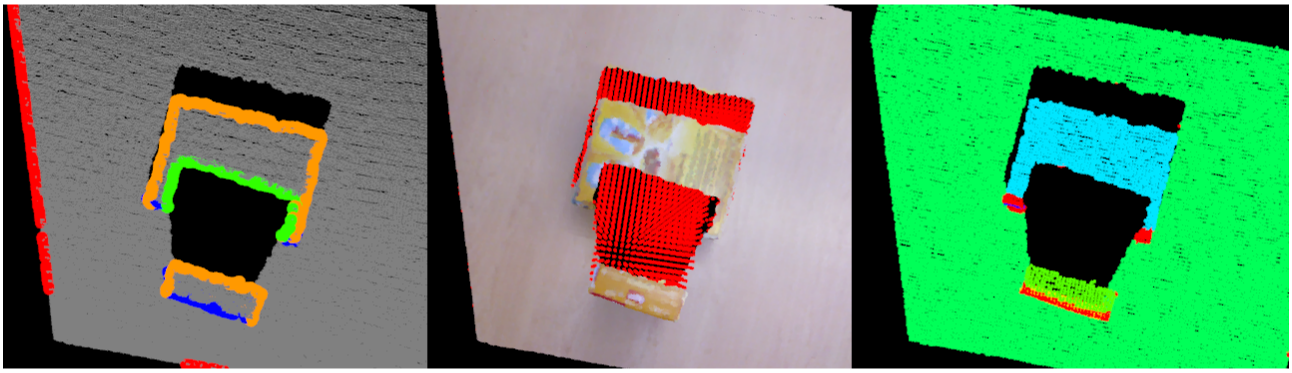


Figure 2: From left to right: different types of edges computed on the point cloud, occlusion volume and smooth superpixels.

expanded bounding box b_ϵ:

$$\delta_o(b) = \frac{|\varepsilon(b)|}{|\varepsilon(b_\epsilon)|} \qquad (2)$$

It is worth noting that this term is close to 1 when the surroundings do not contain edges and decreases linearly to 0 otherwise.

3.3. Smooth superpixels straddling

A third cue on which our objectness term relies aims at penalizing bounding boxes that intersect smooth surface segments (*superpixels*), as this is usually the indication that the bounding box does not contain entirely one object. Indeed, since one common assumption is that superpixels do not to straddle different objects, bounding boxes intersecting a superpixel are penalized. The smooth segments are obtained performing an over-segmentation of $\mathcal{S}$ based on point proximity and surface curvature smoothness. The right part of Figure 2 shows the results of such over-segmentation stage. Observe how non smooth regions are all assigned the same label 0 (depicted in red).

Let $|p(b)|$ be the number of points inside a bounding box b with a superpixel label different than 0, $|p(s)|$ the number of points in $\mathcal{S}$ assigned to a superpixel s and $|p(b \cap s)|$ the number of points belonging to s and within b; the third cue $\delta_l(b, s)$ relative to a single superpixel s is then defined as follows:

$$\delta_l(b, s) = \frac{|p(b \cap s)|^2}{|p(s)|} \qquad (3)$$

The final term $\delta_l(b)$ relative to a bounding box b and all its enclosed superpixels can then be obtained as follows:

$$\delta_l(b) = \frac{\sum\limits_{s \in \Omega} \delta_l(b, s)}{|p(b)|} \qquad (4)$$

where Ω is the set of superpixels extracted from $\mathcal{S}$.

3.4. Free space

A final cue taken under consideration regards the free space within a bounding box aim at favoring bounding boxes that tightly enclose the object of interest. Let $|p_{\mathcal{V}_\mathcal{O}}(b)|$ be the number of occluded voxels inside a bounding box b computed by means of the occluded volume $\mathcal{V}_\mathcal{O}$, and let $V(b)$ be the volume of b; the free space cue is then defined as follows:

$$\delta_f(b) = \frac{V(b) - |p(b)| - |p_{\mathcal{V}_\mathcal{O}}(b)|}{a(b)} \qquad (5)$$

3.5. 3D Objectness measure

Given the aforementioned cues, we can thus define the objectness measure for a bounding box b by weighted sum of the previously introduced cues:

$$\delta(b) = w_i \cdot \delta_i(b) + w_o \cdot \delta_o(b) \cdot \delta_l(b) + w_f \cdot \delta_f(b) \qquad (6)$$

As it can be seen, eq. 6 includes also a feature combination aimed at dimensionality reduction of the weights ($\delta_o(b)$ being multiplied by $\delta_l(b)$), this being motivated empirically. Instead of a heuristic measure like the one in eq. 6, a learning approach for the different weights and combination of cues is desired. We leave a more grounded approach outside of the scope of the paper and will address it in future work.

4. Point cloud segmentation

This section details how the objectness measure presented in the previous section can be successfully

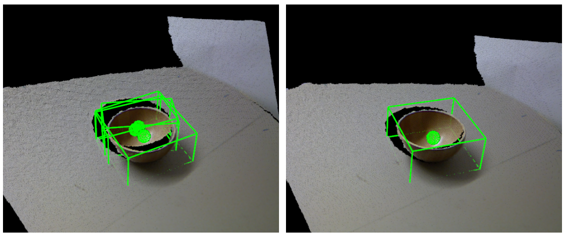


Figure 3: Left: Remaining bounding boxes after the post-processing stage in SubSection 4.1. Right: the final bounding box selected by the method presented in Section 4.2 that simultaneously considers the interaction between all hypotheses to find out a consistent segmentation of the scene.

applied for the task of point cloud segmentation. We carry out this by means of three successive steps: (i) we use the measure itself to filter out bounding boxes below a certain *object* threshold (ii) the remaining bounding boxes are clustered together based on the amount of scene overlap within them, then for each cluster only those with the highest objectness measure are kept, finally (iii) a global cost function is defined over the scene aimed at determining a globally consistent subset of bounding boxes that best segment the scene. The output of the algorithm is thus represented by the set of bounding boxes surviving these three steps or, equivalently, the scene segment associated to each bounding box, where a segment includes all pixels falling within a bounding box.

4.1. Filtering bounding boxes based on objectness

Consider $\mathcal{B} = \{b_1, \cdots, b_N\}$ a set of bounding boxes with an associated objectness score. As a first step, bounding boxes are discarded by thresholding the objectness score.Successively, we group the remaining bounding boxes into a set of clusters $\mathcal{C} = \{c_1, ..., c_m\}$, where each cluster c_i groups together all bounding boxes having their center within the same radius. Within each cluster c_i we analyze the first n_b, sorted by their objectness score, looking for conflicting bounding boxes. We say that two bounding boxes b_i, b_j are in conflict if they share at least 95% of scene points within them (with respect to the bounding box with a higher amount of points). We create a *conflict graph* within each cluster and perform a non-maxima supression based on the objectness measure to keep the best bounding box among those in conflict. This process results in a new set of bounding boxes $\mathcal{B}^* = \{b_1, \cdots, b_n\}$, with usually $n << N$. The left part of Figure 3 shows the bounding boxes after this stage while, on the right, it depicts the finally selected bounding boxes by means of the final post-processing stage, presented in the next section.

4.2. Global hypotheses selection

Here we provide a framework for establishing the most plausible configuration of salient objects in the current scene under evaluation. The problem can be formalized as follows. We start from a set of n object hypotheses, $\mathcal{B}^* = \{b_1, \cdots, b_n\}$, represented by the bounding boxes that survived the filtering step described in Section 4.1.

We adopt the framework (and notation) proposed in [3] to optimize the problem of finding the best configuration of plausible hypotheses simultaneously present in $\mathcal{S}$. Specifically, a cost function is defined over the solution space defined by the set of boolean variables $\mathcal{X} = \{x_0, \cdots, x_n\}$ having the same cardinality as $\mathcal{B}^*$, with each $x_i \in \mathbb{B} = \{0, 1\}$ indicating whether the corresponding hypothesis $h_i \in \mathcal{B}^*$ is dismissed/validated (i.e. $x_i = 0/1$). Hence, the problem can be formulated as finding the best configuration that minimizes a *cost* function expressed as $\mathfrak{F}(\mathcal{X}) : \mathbb{B}^n \to \mathbb{R}$, $\mathbb{B}^n$ being the solution space, of cardinality 2^n:

$$\tilde{\mathcal{X}} = \underset{\mathcal{X} \in \mathbb{B}^n}{\operatorname{argmin}} \{ \mathfrak{F}(\mathcal{X}) \} \tag{7}$$

where

$$\mathfrak{F}(\mathcal{X}) = \sum_{i=1}^{n} \delta_f(b_i) \cdot x_i + \lambda \sum_{p \in \mathcal{S}'} \omega_{\mathcal{X}}(p) \tag{8}$$

As it can be seen from (8), the cost function we aim at minimizing is composed by two terms weighted by a regularizing parameter λ. The left-hand term aims at enforcing tight bounding boxes around the objects, and thus penalizes the free space (through term $\delta_f(b_i)$) of the currently activated bounding box hypotheses within configuration $\mathcal{X}$. As for the right-hand term, it is a sum over all points of the scene surface $\mathcal{S}'$: for each point, a weight $\omega_{\mathcal{X}}(p)$ is thereby associated which enforces instead several cues penalizing invalid combinations of active bounding boxes over p in the current configuration $\mathcal{X}$. $\mathcal{S}'$ represents the initial scene $\mathcal{S}$ downsampled to a lower resolution (for efficiency reasons) after removing points on $\mathcal{P}$ or below it.

To define this weight, we first have to introduce a new term, $\kappa(p, b)$, which takes the value 1 when the point p is within the bounding box b, 0 otherwise. On top of the definition of $\kappa(p, b)$, we define a term $\kappa(p)_\mathcal{X}$ counting the number of bounding boxes activated within a specific configuration $\mathcal{X}$ that enclose point p:

$$\kappa_\mathcal{X}(p) = \sum_{i=1}^{n} \kappa(p, b_i) \cdot x_i \tag{9}$$

The weight $\omega_\mathcal{X}(p)$ can be thus defined as:

$$\omega_\mathcal{X}(p) = \begin{cases} \sum_{i=1}^{n} \kappa(p, b_i) \cdot x_i, & \kappa_\mathcal{X}(p) > 1 \\ -\sum_{i=1}^{n} \kappa(p, b_i) \cdot x_i \cdot \delta(b_i), & \kappa_\mathcal{X}(p) = 1 \\ \sum_{i=1}^{n} \kappa(p, b_{i,\epsilon}) \cdot x_i, & \kappa_\mathcal{X}(p) = 0 \end{cases} \tag{10}$$

The three conditions included in (10) are relative to three different cues being simultaneously enforced by the proposed cost term. In the first condition (case i), $\kappa_\mathcal{X}(p) > 1$), point p introduces a penalty due to the fact that it being enclosed by more than one bounding box (*multiple assignment*). The penalty is proportional to the number of bounding boxes enclosing p. As for the second condition (case ii), $\kappa_\mathcal{X}(p) = 1$), the cost is being penalized by the objectness measure associated to the unique bounding box that encloses p, as we aim at retaining hypotheses characterized by high objectness. Finally, as for the third condition (case iii), $\kappa_\mathcal{X}(p) = 0$), if a point is not enclosed by any bounding box, it adds a penalty to all active hypotheses for which it falls in their proximity. This final cue is computed by means of an expanded bounding box b_ϵ as done in (2), and tends to penalize a bounding box if it has points lying in its surroundings that are not explained by any other active hypotheses, this being usually a sign of a not good enclosure of the object.

To find a minimum for the cost function $\mathfrak{F}$ we deploy Simulated Annealing[7], a typical meta-heuristic algorithm used for finding approximated solutions of non-linear pseudo-boolean programming problems.

5. Experimental evaluation

In order to assess the performance of the 3D objectness measure as well as of the proposed segmentation approach presented in Section 4, we have performed an evaluation regarding segmentation accuracy on the publicly available Willow ICRA Challenge dataset [2] containing 434 object instances lying on a table. The dataset contains pixelwise annotated ground-truth segmentation and allows us to evaluate over- and under-segmentation. It contains typical household objects such as cereal boxes, food cans, detergent bottles, books, etc. (see Figure 4-(d)). Figure 1 show some scenes from the dataset.

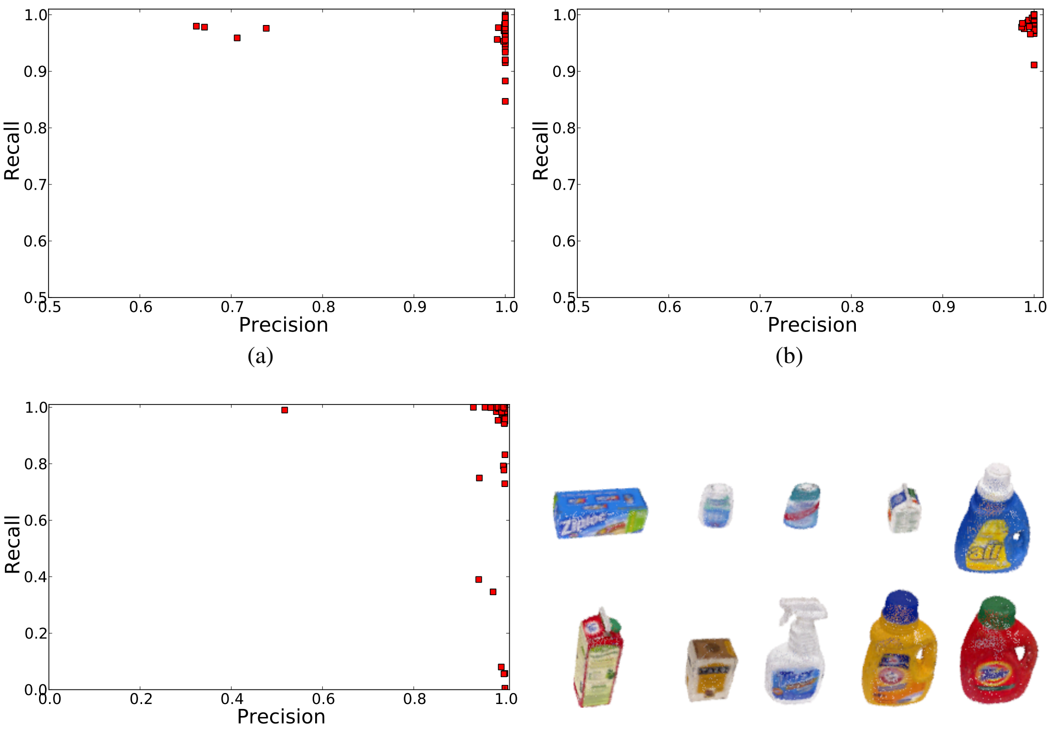
We compare the performance of our method with the segmentation method based on [6]; a simple but highly efficient two step strategy: (i) multi-plane segmentation of the scene and (ii) connected component clustering of points above any detected plane. To efficiently compute planar regions in a scene, it uses a connected components strategy where neighboring pixels are considered to be in the same component (planar region in this case) if the dot product of their normals and the Euclidean distance between the points are within a certain range. The planar regions found are further analyzed in order to merge regions that share the same planar model and were not detected during the first stage due to the constrained 4 neighborhood search. The second step performs similarly to the first step, and groups points (without taking into consideration the points belonging to the detected planes) in the same component if their Euclidean distance is smaller than τ. The resulting components form the segmentation hypotheses. Such a segmentation strategy assumes that the objects will lie on a planar surface and that points belonging to different objects lie farther than τ. Hereinafter, we will refer to this method as *MPS*.

Additionally, we carried out an experiment to evaluate solely the objectness measure. To do so, we computed, on the same dataset, the Precision and Recall values for the bounding box with highest objectness score. In this case, we are interested in assessing how often the bounding box with highest score completely contains a ground truth object without including other objects or part of the background.

5.1. Results and discussion

Figure 4-(a) and -(b) compare respectively Precision and Recall results yielded by *MPS* and the proposed method for the task of scene segmentation. Each point in the scatter plot represents one scene in the dataset. The Precision and Recall values are computed for each scene by averaging the respective

[2] The whole dataset with annotated segmentation labels can be downloaded from `http://svn.pointclouds.org/data/ICRA_willow_challenge_segmentation_gt/`



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

Figure 4: Evaluation on the Willow ICRA Challenge dataset (Precision vs Recall per scene) (a) segmentation results with *MPS* and (b) results with the proposed pipeline. (c) Precision and recall values for the bounding box with highest objectness score. (d) 10 of the 35 objects in the dataset.

values relative to each ground-truth object present in the scene. A low Precision value indicates that the object was undersegmented, while a low Recall indicates that the object was oversegmented. Observe how *MPS* presents undersegmentation on 4 scenes were objects are touching each other as well as oversegmentation on several other scenes caused by self-occlusions or missing data. Overall, the proposed approach outperforms *MPS*, with an average Precision/Recall of 99.9% versus 99.1%.

Figure 4-(c) shows the Precision and Recall results obtained by considering only the bounding box with highest objectness score. We can observe how just a single bounding box with high objectness resulted in an undersegmentation of a scene. On the other hand, the best bounding box is relatively often presenting oversegmentation, yielding Recall values below 0.9. The scatter points with very low Recall values ($< 10\%$) appear when the best bounding box encloses only background (in some scenes, the hand of the person setting up the dataset is visible and within the VoI but annotated as background).

By analyzing Figure 4-(b) and -(c) simultaneously we can note that even on situations where the bounding boxes with higher objectness were causing oversegmentation (Recall values below 0.9), the segmentation method in Section 4 was ultimately selecting other bounding boxes providing a more pleasant and consistent configuration. The same applying for the undersegmentation case, indicates that simultaneously analyzing nearby bounding boxes allows to overcome some errors caused by individual local decisions.

6. Conclusion and future work

In the context of this paper, we have presented several cues derived from a 3D point cloud to evaluate how likely it is for a closed region in 3D space to enclose completely a single object. The cues have been combined in a preliminar *objectness* measure formulation that has shown great potential during the experiments. We have also presented a framework for scene segmentation based on the *objectness* measure as well as other physical constraints being able to find a plausible segmentation of table-top scenarios even under challenging situations where objects

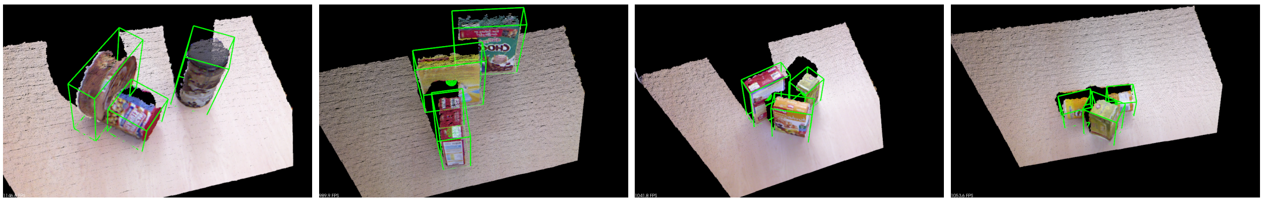


Figure 5: Some qualitative results with the presented method. Observe on the most right image in the figure a failure case where the cylinder is splitted into two bounding boxes. These scenes belong to the *Object Segmentation Database* `http://www.acin.tuwien.ac.at/?id=289`.

are touching each other.

Based on the encouraging results obtained in this initial work as well as the observed limitations of the methods, there exist several directions that ought to be explored in the future. As already pointed out in Section 3.5, a more grounded approach for the combination of cues needs to be investigated as well as additional cues that might help solving even more challenging scenes. Another direction of research aims at reducing the computational complexity of the method in order to be able to explore the additional degrees of freedom that were ignored in the scope of this work (especially, we would like to allow objects to be on top of each other removing the table-top constraint). In this direction, we would like to explore bottom-up strategies to infer promising subspaces where the objectness measure will be evaluated. This strategy would allow to replace the current exhaustive enumeration resulting in a much lower complexity even when removing some or all of the constraints currently used.

Acknowledgements

The research leading to these results has received funding from the Doctoral College on Computational Perception at TU Vienna and the Austrian Science Foundation under grant agreement No. I513-N23.

References

[1] A. Aldoma, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, M. Vincze, and G. Bradski. CAD-Model Recognition and 6DOF Pose Estimation Using 3D Cues. In *Workshop: 3rd IEEE Workshop on 3D Representation and Recognition, ICCV*, 2011. 1, 2

[2] A. Aldoma, F. Tombari, R. Rusu, and M. Vincze. Our-cvfh: Oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation. In *Joint DAGM-OAGM Pattern Recognition Symposium*, 2012. 1, 2

[3] A. Aldoma, F. Tombari, L. D. Stefano, and M. Vincze. A global hypotheses verification method for 3d object recognition. In *Proc. European Conf. on Computer Vision*, 2012. 2, 5

[4] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR'10*, pages 73–80, 2010. 1, 2, 3

[5] K. G. Derpanis. Integral image-based representations. Technical report, 2007. 2

[6] D. Holz, A. J. B. Trevor, M. Dixon, S. Gedikli, and R. B. Rusu. Fast segmentation of rgb-d images for semantic scene understanding. In *ICRA 2012 Workshop on Semantic Perception and Mapping for Knowledge-enabled Service Robotics*. 1, 6

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, (4598), 1983. 6

[8] A. K. Mishra, A. Shrivastava, and Y. Aloimonos. Segmenting ”simple” objects using rgb-d. In *ICRA*, pages 4406–4413. IEEE, 2012. 1

[9] A. Richtsfeld, T. Mörwald, J. Prankl, M. Zillich, and M. Vincze. Segmentation of Unknown Objects in Indoor Environments. In *IROS*, 2012. 1

[10] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 10/2010 2010. 2

[11] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of Histograms for local surface description. In *Proc. 11th European Conference on Computer Vision (ECCV 10)*, 2010. 2