Cluttered Scene Segmentation Using the Symmetry Constraint

Aleksandrs Ecins, Cornelia Fermüller and Yiannis Aloimonos

Abstract— Although modern object segmentation algorithms can deal with isolated objects in simple scenes, segmenting non-convex objects in cluttered environments remains a challenging task. We introduce a novel approach for segmenting unknown objects in partial 3D pointclouds that utilizes the powerful concept of symmetry. First, 3D bilateral symmetries in the scene are detected efficiently by extracting and matching surface normal edge curves in the pointcloud. Symmetry hypotheses are then used to initialize a segmentation process that finds points of the scene that are consistent with each of the detected symmetries. We evaluate our approach on a dataset of 3D pointcloud scans of tabletop scenes. We demonstrate that the use of the symmetry constraint enables our approach to correctly segment objects in challenging configurations and to outperform current state-of-the-art approaches.

I. INTRODUCTION

Rapid advances in robotic technology are bringing robots out of the controlled environments of assembly lines and factories into the unstructured and unpredictable "real-world" workspaces of human beings. One of the prerequisites for operating in such environments is the ability to segment previously unseen objects in cluttered scenes. This remains a challenging task due to the lack of prior information about the shape and pose of the object as well as due to occlusions in clutter.

In this paper we introduce a novel approach to object segmentation in partial 3D pointclouds and apply it to heavily cluttered tabletop scenes. At the core of our approach is the observation that the three dimensional shape of common objects is bilaterally symmetric. Although this assumption might seem restrictive at first, it turns out that very few man-made objects do not have this property [1]. Mugs, chairs, boxes, bottles, computer screens all have at least one bilateral symmetry plane. Moreover, it was argued that symmetry is so prevalent in nature that the visual systems of humans and animals use symmetry both to guide visual attention [2] and as a cue for figure-ground segmentation [3]. Our proposed algorithm consists of two tightly coupled steps. In the first step, candidate 3D bilateral symmetries are detected by extracting and matching surface normal edge curves of the pointcloud. Since the matching is done at curve feature level it is more efficient than the traditional approach of matching individual oriented points. In the second step these candidates are used to initialize multiple figure ground segmentations. The goal of each segmentation is to find points in the scene that belong together according to the grouping principles of proximity and convexity, and

*The authors are with the Department of Computer Science, University of Maryland, College Park, MD 20742, USA {aecins, fer, yiannis}@umiacs.umd.edu

(a) Input pointcloud

(b) Our segmentation

Fig. 1: Example of a scene segmetned using our approach. Color information is not used in the segmentation process and is only used for visualization.

at the same time are geometrically consistent with a given symmetry candidate. The key insight to our segmentation approach is that symmetry hypotheses that correspond to one of the objects in the scene are consistent with all of the observed points of that object, while incorrect symmetries are compatible with an unorganized set of points that can be discarded in later processing stages.

To evaluate our method, we have collected a novel dataset of tabletop scenes. Unlike most such datasets which contain single RGB-D frames, we reconstruct high-quality point-clouds of the scenes observed from multiple points using an open source implementation of the KinectFusion algorithm [4]. Additionally, in order to keep track of occlusions, we construct a volumetric occupancy map for each of the scenes [5]. Evaluation on this dataset shows that the use of the symmetry constraint allows our approach to correctly and accurately segment complex scenes containing non-convex objects as well as multiple objects in stacked or touching configurations while approaches that rely on convexity as the primary grouping principle struggle in such scenarios.

The rest of the paper is organized as follows. We begin by giving an overview of current approaches to object segmentation. Section III presents our approach to finding potential 3D symmetry plane candidates in the scene. In section IV we

describe the details of our segmentation method. In section V we show how to improve segmentation results by refining the initial symmetries. In section VI we show how to filter out incorrect segments based on a number of "objectness" measures. The paper concludes with an evaluation procedure and results in Section VII.

II. RELATED WORK

Image segmentation is one of the fundamental problems in Computer Vision and has been studied extensively. Most of the classical segmentation algorithms were developed for analyzing RGB or grayscale images and use texture, colour and image contour features for grouping [6] [7] [8] [9]. Although suitable for certain tasks, they are not capable of reliably segmenting objects in cluttered environments. Nevertheless, these algorithms serve as a foundation for more recent segmentation approaches which rely on 3D information to overcome many of the limitations of its predecessors.

One such approach is the active segmentation approach of Mishra et al. [10]. Depth and RGB cues are used to find an edge image of the scene, and objects are segmented by finding a closed contour in the edge map around a given fixation point. This method is able to segment compact objects in simple scenes. Additional segmentation cues can be obtained by projecting the depth image into three-dimensional space and analyzing the geometry of the resulting 3D pointcloud. In [11] Richtsfeld et al. presegment the scene by fitting planes and NURBS to the pointcloud data and then merge the resulting segments into object hypotheses. The likelihood of two segments belonging to the same object is computed by training an SVM classifier on a number of appearance and geometrical features. The use of robust shape primitives in the presegmentation step allows this method to deal with compact objects in stacked configurations.

Several methods were proposed that explicitly model the assumption that objects tend to be convex in their shape. Stein et al. [12] use surface normals to oversegment the pointcloud into supervoxels and estimate their adjacency using the distance between their 3D centroids. Pairs of adjacent supervoxels are classified as either convex or concave based on the relative position and surface normal orientation difference of their centroids. Supervoxels are then merged to find components that are enclosed by convex edges. A favorable property of the algorithm is that non-convex objects tend to get divided into their convex parts. In another work Karpathy et al. find objects in 3D meshes of indoor scenes [13]. The input mesh is oversegmented using an adaptation of Felzenswalb's algorithm [6] producing a set of overlapping object hypotheses. Individual hypotheses are then classified as object or non-object based on a number of features including compactness, convexity, smoothness and symmetry. Zheng et al. proposed to reason about physical stability of objects in the scene [14]. Their method estimates the volumetric extent of segments extracted from pointcloud data and then combines them into configurations that are s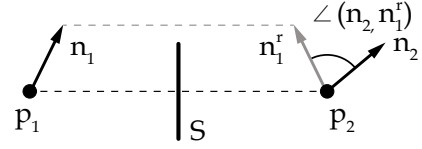table under the gravity constraint. This method is capable of correctly segmenting non-convex objects, but fails in situations when volumetric reconstruction is unreliable.

Fig. 2: Symmetry detection from a pair of oriented points.

Symmetry has been utilized as an attention mechanism to detect potential object locations in the scene. In [15] Koostra et al. generate 2D saliency maps by finding symmetric configurations of image gradients and show that these maps can be used to guide the segmentation process. Potapova et al. [16] showed that higher quality saliency maps can be obtained by extracting locally symmetric depth image patches. More recently Teo et al. [17] proposed to use symmetry both as an object detector and as a constraint in the segmentation process. Curved bilateral symmetries are detected and are used to setup a foreground segmentation that forces the segments to be symmetric with respect to the detected symmetry curves. The idea of using symmetry both as an attention and grouping mechanism is similar to our approach. However, unlike [17] which operates on single 2D images, our approach works on multiple view 3D pointclouds, which enables in to produce accurate results even in extremely cluttered scenes.

III. SYMMETRY DETECTION

Bilateral symmetry detection in 3D data has been studied extensively in the fields of Computer Vision, Robotics and Graphics [18]. Most of the proposed methods rely on the idea of finding symmetric correspondences between oriented points. A bilateral symmetry plane $S = \{n^s, d^s\}$ described by a normal n^s and distance to the origin d^s reflects a point p associated with surface normal vector n to a reflected point p^r, n^r such that:

$$p^r = p - 2n^s(p \cdot n^s - d) \tag{1}$$
$$n^r = n - 2n^s(n^s \cdot n) \tag{2}$$

The coordinates of any two points in space p_1 and p_2 uniquely define a bilateral symmetry plane that is perpendicular to the line connecting the points and contains the midpoint between them. The parameters of this plane are:

$$n^s = \frac{p_1 - p_2}{\|p_1 - p_2\|} \tag{3}$$
$$d^s = \frac{(p_2 - p_1) \cdot n^s}{2} \tag{4}$$

The matching score of the symmetric correspondence between p_1 and p_2 can be measured by the angular difference between the normal of one of the points and the reflected normal of the other point:

$$M_S(p_1, p_2) = \angle(n_1, n_2^r) \tag{5}$$

Thus a naive approach to discovering symmetries in a pointcloud is to find every pair of points that forms a valid
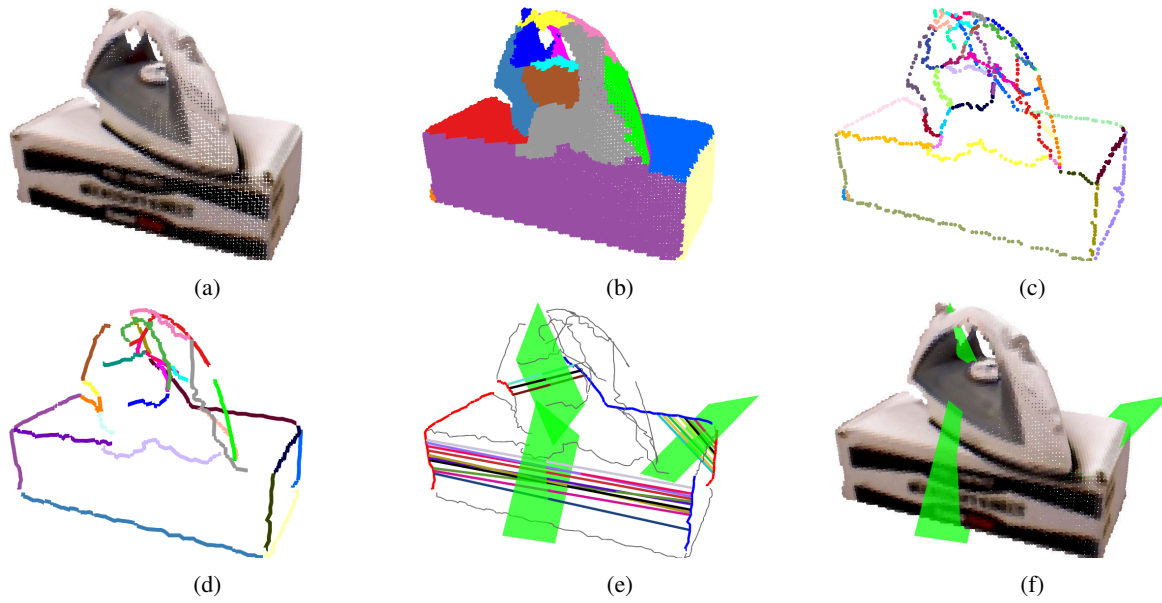
Fig. 3: Symmetry detection. (a) Input pointcoloud. (b) Overegmentation. (c) Segment boundary points. (d) Linked edge curves. (e) and (f) Symmetry correspondences and resulting symmetry hypotheses. Only 3 of the 61 symmetry hypotheses detected for this scene are shown here.

symmetry correspondence and then filter the set of symmetry hypotheses to recover the dominant symmetries. However this is computationally prohibitive since the number of point pairs that need to be checked is quadratic in the number of points in the pointcloud. An alternative is to find correspondences between distinctive features of the pointcloud that are represented by a smaller number of points but at the same time exhibit the same symmetries as the original pointcloud. One candidate for such features are *surface normal edges* i.e. points where the surface normal direction changes abruptly. By eliminating points belonging to smoothly varying surfaces, normal edges can be represented by a fraction of the points of the original pointcloud while preserving most of the geometrical information, allowing symmetries to be detected efficiently and accurately.

Our strategy for extracting normal edges is similar to that presented in [19] which utilizes the duality between segmentation and edge detection. We oversegment the pointcloud into patches whose borders are likely to lie along normal edges and then find the boundaries between them. We use the algorithm of Papon et al. [20] that segments pointclouds into supervoxels with high boundary adherence. To avoid extracting boundaries between supervoxels that belong to the same flat surface, we merge all adjacent supervoxels that have collinear surface normals (Figure 3b). Superovxel boundary points are found by extracting points which have at least one neighbour that belongs to a different supervoxel (Figure 3c). Boundary points are then linked together using a Minimum Spanning Tree algorithm to form a set of disjoint edge curves $\mathbf{C} = \{C_1, ..., C_n\}$. Each curve $C_i = (p_{C_i}^1, ..., p_{C_i}^m)$ is represented by an ordered set of points and describes a surface normal edge in the pointcloud. (Figure 3d).

Once the edge curves are extracted, pointcloud symmetries can be recovered by finding symmetrical pairs of edge curves. Given a pair of edge curves C_i, C_j we exhaustively search for symmetrical correspondences between their individual points $\{p_{C_i}, p_{C_j}\}$ using the point matching approach described above, with the difference that the curve tangent direction is used as orientation vector instead of the surface normals. Matches are filtered by removing correspondences with a match score higher than a threshold $M_S(p_{C_i}, p_{C_j}) > 6°$ and enforcing a one-to-one correspondence relationship. This results in a set of noisy candidates for the global symmetry plane aligning the two curves. To recover the global symmetry plane we take an approach similar to the one proposed in [21]. Symmetrical candidates are treated as samples of the probability density function of the global symmetry plane. The problem of recovering the global symmetry plane becomes equivalent to clustering these candidates. To do the clustering we represent the candidates with points in three dimensional space by weighting the symmetry plane normal by the distance to the origin i.e. n^s/d^s. We then use mean shift clustering [22] to find the dominant mode of the distribution and use its maximum as the symmetry plane aligning the two curves. This procedure is run on every pair of edge curves resulting in a set of 3D bilateral symmetry hypotheses for the input pointcloud $\mathbf{S} = \{S_1, ..., S_k\}$ (Figure 3e). It is important to note that the resulting set of hypotheses will inevitably contain a large number of false hypotheses. This however does not pose a problem since the subsequent stages of the processing pipeline will discard most of the erroneous symmetries and only require that the initial symmetry hypothesis set contains symmetries that are close enough to all of the true symmetries of the pointcloud.
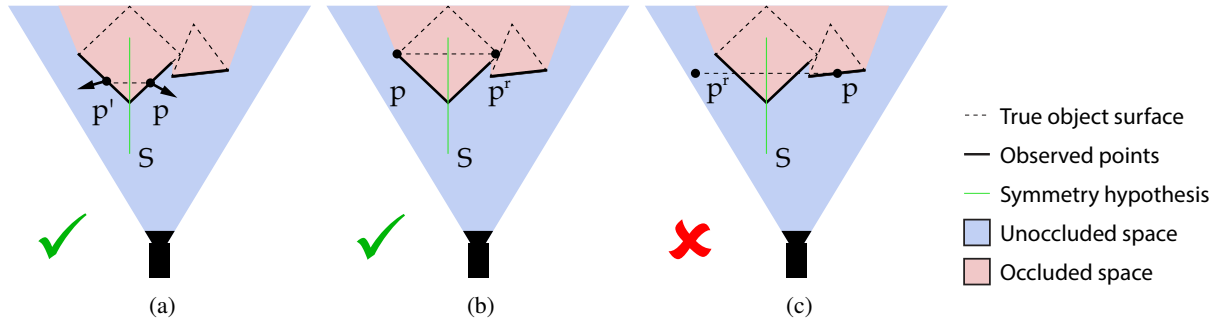
Fig. 4: Symmetry consistency analysis. A square and a triangle are observed by the camera. p^r denotes the reflection of point p and p' denotes the symmetric neighbor of p. (a) p reflects to its symmetric neighbor. (b) p reflects to occluded space. (c) p reflects to unoccluded space. Note that all of the observed points belonging to the square and none of the points of the triangle are consistent with symmetry S.

IV. SYMMETRY SEGMENTATION

The goal of the segmentation step is to find object hypotheses i.e. subsets of points in the pointcloud that are consistent with the detected bilateral symmetries and at the same time respect the basic grouping laws of proximity and convexity. Consider the set of visible surface points $\mathcal{P}$. To check if a point $p \in \mathcal{P}$ is consistent with a symmetry hypothesis S we analyze its reflection [23]. We distinguish between three different cases. If p is reflected to another point of the pointcloud p', we call them symmetric neighbors and they are both considered to be supporting the symmetry S (Figure 4a). If on the other hand it reflects into occluded space it does not support the symmetry but is still compatible with it (Figure 4b). Finally if a point reflects to unoccluded space it violates the symmetry S (Figure 4c).

To segment all of the objects in the scene we setup multiple graph based foreground segmentation problems. A graph is constructed for each of the symmetry hypotheses where nodes correspond to the points of the pointcloud and edges are established between adjacent points of the pointcloud as well as between the symmetrical neighbours under a given symmetry. Let $\mathcal{L} = \{fg, bg\}$ be labels corresponding to object and background respectively. Object hypotheses are segmented by finding a labeling f that assigns a label $f_p \in \mathcal{L}$ to all points in the pointcloud, such that the following energy functional is minimized:

$$
\begin{aligned}
E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) \quad & + \\
\lambda_{smooth} \sum_{\{p_1,p_2\} \in \mathcal{N}_{smooth}} V^{smooth}_{p_1,p_2} \cdot \delta(f_{p_1} \neq f_{p_2}) \quad & + \\
\lambda_{sym} \sum_{\{p,p'\} \in \mathcal{N}_{sym}} V^{sym}_{p,p'} \cdot \delta(f_p \neq f'_p) &
\end{aligned}
\tag{6}
$$

where $\delta(\cdot)$ denotes an indicator function. The first term is the symmetry consistency term $D_p(f_p)$ that ensures that the points consistent with the current symmetry hypothesis are labeled as foreground. $V^{smooth}_{p_1,p_2}$ is the smoothness term defined between neighbouring points of the pointcloud. It forces segment boundaries to lie along the surface normal

edges and not across flat surfaces. Finally $V^{sym}_{p,p'}$ is the symmetry term that sums up the cost of assigning different labels to symmetric points in the pointcloud. λ_{smooth} and λ_{sym} determine the influence of the smoothness and symmetry terms. In our experiments we set both of these weights to 2. Once the graphs are constructed they are optimized using the graph-cuts algorithm [24]. To speed up graph inference we reduce the number of points in the scene pointcloud by downsampling it with a voxel grid filter with voxel size of 0.01 metres. We will now discuss how we compute each of the terms in the graph.

A. Symmetry consistency term

Given a symmetry hypothesis S, a point p is reflected to p^r. We can find the symmetrical neighbour of p by searching for the nearest neighbour p' of the reflected point and checking the distance between them. If it is smaller than a threshold $\|p^r - p'\| < d_{max}$, point p is likely to belong to the object and hence its background weight is set to 0. Foreground weight is set according to the angle between the reflected normal and the symmetrical neighbour normal:

$$
D_p(fg) = 1 - \frac{\angle(n^r, n') - \alpha_{min}}{\alpha_{max} - \alpha_{min}}
\tag{7}
$$

$$
D_p(bg) = 0
\tag{8}
$$

On the other hand, if the distance to the nearest neighbour is greater than d_{max}, point p has no symmetrical neighbours and its foreground weight is set to 0. The background weight is set to 0 if the reflected point occupies occluded space. If it reflects to unoccluded space its background weight is set proportionally to the distance to the nearest point:

$$
D_p(fg) = 0
\tag{9}
$$

$$
D_p(bg) = \begin{cases} \dfrac{\|p^r - p'\| - d_{max}}{d_{buf}} & \text{if } p^r \in \text{unoccluded space} \\ 0 & \text{otherwise} \end{cases}
\tag{10}
$$

The following constant values are used: $d_{max} = 0.01$, $\alpha_{min} = 15°$, $\alpha_{max} = 45°$ and $d_{buf} = 0.03$.

B. Smoothness term

Point adjacency in the pointcloud $\mathcal{N}_{smooth}$ is estimated by connecting every point to 5 of its nearest neigbours. Smoothness weights are set between every pair of neighbouring points. One viable option is to set the edge weights according to the angle between the surface normals of neighbouring points. A smaller angle indicates that points belong to the same surface while a large angle suggests the existence of a surface normal edge between the points. This approach can be improved by modulating the weights based on the convexity criterion. This modification captures the intuition that objects tend to be convex while boundaries between two objects on top of each other tend to be concave. Given two points p_1 and p_2 we define them to be in a convex arrangement if $n_1 \cdot (p_1 - p_2) > 0$. The binary weight between them is set as:

$$V_{p_1,p_2}^{smooth} = \begin{cases} \exp\left(-\dfrac{\angle(n_1, n_2)}{\sigma_{convex}}\right) & \text{if } n_1 \cdot (p_1 - p_2) > 0 \\ \exp\left(-\dfrac{\angle(n_1, n_2)}{\sigma_{concave}}\right) & otherwise \end{cases} \quad (11)$$

where $\sigma_{convex} = 360°$ and $\sigma_{concave} = 30°$.

C. Symmetry term

Symmetrical neighbour edges are established between all points which are reflected to existing points of the pointcloud i.e. $\mathcal{N}_{sym} = \{p, p' \mid \|p^r - p'\| < d_{max}\}$. Once again the weights are set based on the difference in their normal angles:

$$V_{p,p'}^{sym} = 1 - \frac{\angle(n, n') - \alpha_{min}}{\alpha_{max} - \alpha_{min}} \quad (12)$$

The values of the constants are the same as in the symmetry consistency term.

V. REFINEMENT

The output of the segmentation step is a set of object hypotheses $\mathbf{H} = \{H_1, ..., H_n\}$ associated with bilateral symmetry planes $\mathbf{S} = \{S_1, ..., S_n\}$ used to generate them. Importantly, most of the incorrect symmetry planes are likely to result in empty object hypotheses which can be immediately discarded. This is due to the fact that incorrect symmetry hypotheses are not compatible with an object-like sets of points in the observed scene pointcloud. However, noise in the detected symmetry planes can result in incomplete object hypotheses even if they are close to the correct symmetries of the scene (Figure **??**). These types of errors can be corrected by aligning the original object hypotheses with their reflected versions. Points of hypothesis H are reflected around the corresponding symmetry plane S and then aligned back to H using the ICP algorithm [25] (Figure 5c). To ensure fast and accurate convergence we use the point-to-plane distance metric for correspondence estimation and force correspondences to be one-to-one. A refined symmetry estimate $\bar{S}$ is then recovered by applying the mean shift filtering technique used in the symmetry detection step on the correspondences between the points of the original and the reflected pointclouds. Finally the segmentation procedure
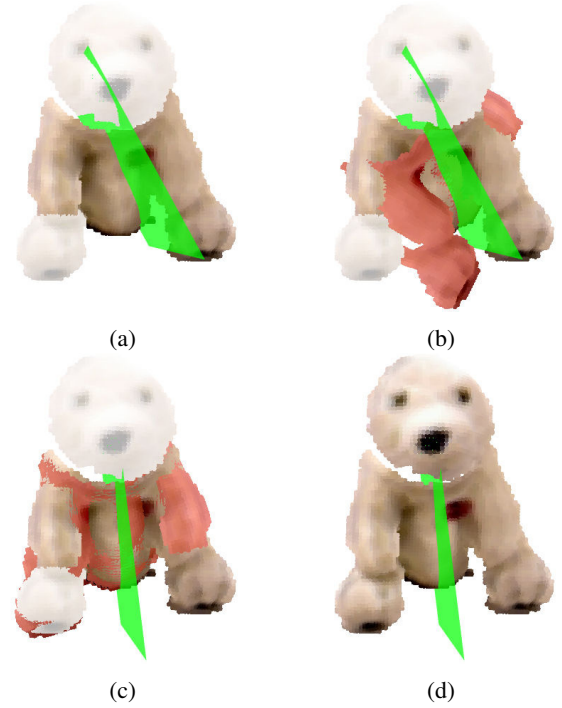

(a)　　　　　　　　(b)

(c)　　　　　　　　(d)

Fig. 5: Segmentation refinement process. (a) Initial symmetry is not precise, leading to an incomplete object hypothesis. (b) Initial object hypothesis is reflected around the symmetry (shown in red). (c) They are aligned and a more accurate symmetry estimate is recovered from aligned pointclouds. (d) Repeating the segmentation process with the refined symmetry returns a correct object hypothesis.

described above is run again using the refined symmetry hypothesis $\bar{S}$ (Figure 5d). This procedure is applied to all of the non-empty object hypotheses resulting in the refined set of hypotheses $\bar{\mathbf{H}} = \{\bar{H}_1, ..., \bar{H}_m\}$.

VI. SEGMENTATION FILTERING

In order to reduce the number of false object hypotheses we employ a filtering process based on a number of "objectness" measures. These measures are used to rate each hypothesis on how likely is it to correspond to an object in the scene.

Smoothness score. Object hypotheses should be smooth i.e. their boundaries should not cross flat surfaces. We measure the smoothness of a hypothesis H by counting the average number of smooth links between adjacent points of the pointcloud that are broken by the segmentation. To make sure that surface normal edges do not contribute to the score we only count links for which the angle between point normals is smaller than a threshold. Let $\mathcal{B}_H = \{p_i, p_j \mid p_i \in H, p_j \in \mathcal{P}/H, \{p_i, p_j\} \in \mathcal{N}_{smooth}\}$ be the set of links between neighbouring points in the pointcloud that were broken by the object hypothesis H. The smoothness score is calculated as:

$$Smooth(H) = \frac{1}{|H|} \sum_{\{p_i,p_j\} \in \mathcal{B}_H} \delta(\angle(n_i, n_j) < \alpha_{smooth}) \quad (13)$$

Fig. 6: Example of a multiple view pointcloud of a scene from the dataset shown from different viewpoints.

Occlusion score. All of the points of a valid object hypothesis should reflect to the occluded space. We evaluate this property by calculating the fraction of the hypothesis points that fullfill this requirement:

$$Occl(H) = \frac{1}{|H|} \sum_{p_i \in H} \delta(p_i \in \text{occluded space}) \qquad (14)$$

Object hypotheses that satisfy both $Smooth(H) < s_{smooth}$ and $Occl(H) > s_{occl}$ are considered valid. The rest of the hypotheses are rejected.

VII. EXPERIMENTS AND RESULTS

A. Dataset

To evaluate our method we collected a dataset consisting of 89 scenes of various objects placed on a table. The set of objects included simple objects like boxes as well as non-convex objects such as a teddy bear. Complexity of the scenes varied from single objects to multiple objects put side by side and stacked on top of each other. Scenes were captured by moving a Kinect sensor in approximately $60°$ horizontal arc around the center of the scene. Object points were extracted as points lying above the table plane and their ground truth segmentation masks were manually labeled. An example of a pointcloud from the dataset is shown in Figure 6.

B. Evaluation procedure and methods

Unlike the majority of modern segmentation algorithms which assign a single object label to every point in the pointcloud, our method returns a set of object segments that can be spatially overlapping. Moreover, some points might not get assigned to any segment at all. We believe that this property does not reflect negatively on our approach. We argue that the utility of a segmentation process for a robotic system should be measured not by the number of points in the scene that were assigned a correct object label, but by the number of objects in the scene for which sufficiently accurate segmentation masks were returned. This principle motivates our choice of the evaluation metric. An object in the scene is considered to be segmented correctly if there exists a predicted segment that overlaps sufficiently with its ground truth segmentation mask (intersection over union metric is used to compute the overlap). Recall is calculated as the fraction of the objects in the dataset that were segmented correctly and precision as the number of correctly segmented
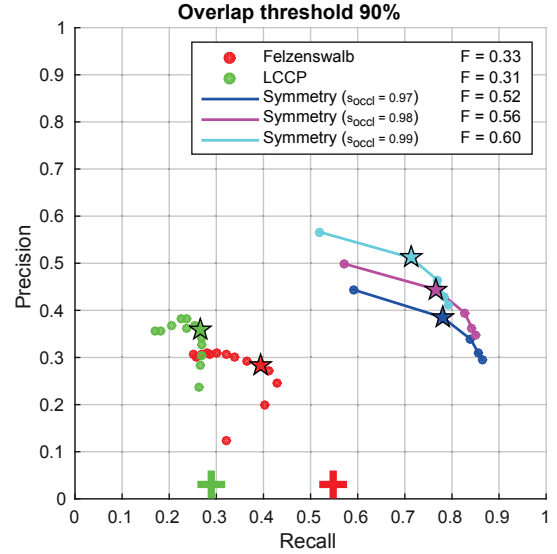

Fig. 7: PR curves for segmentation evaluation. Stars mark the the maximum F-measure for each method. Crosses denote results for hypotheses combined over all thresholds for Felzenswalb and LCCP methods.

objects normalized by the total number of returned segments. Denoting by $\mathbf{H}^T$ the set of ground truth masks for all objects in all of the scenes of a dataset and by $\mathbf{H}^P$ the set of predicted object hypotheses:

$$P = \frac{1}{|\mathbf{H}^P|} \sum_{H_t \in \mathbf{H}^T} \delta(\max_{H_p \in \mathbf{H}^P} O(H_p, H_t) > \sigma_o) \qquad (15)$$

$$R = \frac{1}{|\mathbf{H}^T|} \sum_{H_t \in \mathbf{H}^T} \delta(\max_{H_p \in \mathbf{H}^P} O(H_p, H_t) > \sigma_o) \qquad (16)$$

where

$$O(H_1, H_2) = \frac{H_1 \cap H_2}{H_1 \cup H_2} \qquad (17)$$

and σ_o is the desired overlap score. In our evaluation we use the value of $\sigma_o = 0.9$. We choose such a high overlap requirement since small errors in segmentation can lead to significant errors in later stages of the processing pipeline of a robot (i.e. grasp selection and path planning).

We evaluate the following 3 segmentation algorithms:
Felzenswalb. An adaptation of the Felzenswalb segmentation algorithm for pointcloud data taken from [13]. Segmentation threshold k was varied from 0.5 to 7 in steps of 0.5.
LCCP [12]. A supervoxel-based segmentation with a convexity prior. The following parameters were used: voxel seed resolution $R_{voxel} = 0.005$, supervoxel seed resolution $R_{seed} = 0.04$. Concavity tolerance threshold β_{thresh} was varied between $10°$ and $70°$ in steps of $5°$.
Proposed method. Segmentation parameters were fixed as described above. We show results for three occlusion filtering thresholds $s_{occl} = \{0.97, 0.88, 0.99\}$. For each of them smoothness threshold s_{smooth} was varied from 0.01 to 0.05 in steps of 0.01.

C. Results

Figure 7 shows the precision recall curves for the three algorithms. Felzenswalb and LCCP methods achieve similar performance while the proposed method significantly outperforms both of them. Our method achieves an overall higher precision and recall scores and an F-measure of 0.60. As expected, using stricter filtering thresholds leads to increases in precision at the cost of recall. The maximum F-measure is achieved with $s_{occl} = 0.99$ and $s_{smooth} = 0.02$. Remarkably, our method achieves a very high recall of 86% when using the least aggressive filtering setting ($s_{occl} = 0.97$, $s_{smooth} = 0.05$). This means that even though our algorithm generates a number of false positives, it produces very accurate object segmentation masks for almost all of the objects in the dataset. To compare the same property for Felzenswalb and LCCP we combined all of the hypotheses generated by these algorithms at all threshold levels and calculated the precision recall scores. As shown in Table I these methods achieve a maximum recall of 55% and 29% respectively. Even if a perfect filtering technique, capable of removing all false positives and none of the true positives, was used to improve the results of these methods, their performance would still be limited.

Looking at scenes in Figure 8, we can identify two distinct types of object configurations where Felzenswalb and LCCP methods fail:

1) touching objects where the transition between objects is not concave are merged together. In row 1 transition between the speaker and the coffee box is non-concave.
2) objects separated by an occlusion boundary are over-segmented. In row 2 the box is occluded by the milk carton.

In both of these cases convexity criterion is not sufficient to correctly group the points into objects. On the other hand symmetry provides a global object-level grouping principle, which allows our method to correctly segment these scenes.

Algorithm	F-measure	Highest recall
Felzenswalb	0.33	55%
LCCP	0.31	29%
Symmetry ($s_{occl} = 0.97$)	0.52	**86%**
Symmetry ($s_{occl} = 0.98$)	0.56	85%
Symmetry ($s_{occl} = 0.99$)	**0.60**	79%

TABLE I: Segmentation quantitative evaluation.

VIII. CONCLUSIONS AND FURTHER WORK

In this paper we demonstrated how bilateral symmetry can be used as a prior for segmenting objects in 3D point-clouds of cluttered scenes. Symmetry is used both as an attention mechanism that provides seeds for the segmentation process and as a grouping mechanism that helps solve the figure-ground organization problem. Our approach requires no training and relies solely on geometric information to segment scenes. Evaluation on a novel challenging dataset shows that it outperforms current state-of-the-art approaches.

Most importantly, it achieves very high recall scores. In the future work we plan to increase the precision of our approach by in employing more advanced object hypothesis filtering techniques. Furthermore we will extract accurate object level symmetries for the segmented objects and use them to reconstruct the occluded parts of the objects.

REFERENCES

[1] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Symmetry descriptors and 3D shape matching," in *Symposium on Geometry Processing*, Jul. 2004.
[2] C. W. Tyler, *Human symmetry perception and its computational analysis*. Psychology Press, 2003.
[3] J. Driver, G. C. Baylis, and R. D. Rafal, "Preserved figure-ground segregation and symmetry perception in visual neglect," *Nature*, vol. 360, no. 6399, pp. 73–75, 1992.
[4] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 2011, pp. 127–136.
[5] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013.
[6] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.
[7] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000.
[8] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 105–112.
[9] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 309–314, 2004.
[10] A. Mishra, Y. Aloimonos, and C. Fermuller, "Active segmentation for robotics," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 3133–3139.
[11] A. Richtsfeld, T. Morwald, J. Prankl, M. Zillich, and M. Vincze, "Segmentation of unknown objects in indoor environments," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4791–4796.
[12] S. C. Stein, M. Schoeler, J. Papon, and F. Worgotter, "Object partitioning using local convexity," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 304–311.
[13] A. Karpathy, S. Miller, and L. Fei-Fei, "Object discovery in 3d scenes via shape analysis," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2088–2095.
[14] B. Zheng, Y. Zhao, J. C. Yu, K. Ikeuchi, and S.-C. Zhu, "Beyond point clouds: Scene understanding by reasoning geometry and physics," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3127–3134.
[15] G. Kootstra, N. Bergstrom, and D. Kragic, "Fast and automatic detection and segmentation of unknown objects," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*. IEEE, 2010, pp. 442–447.
[16] E. Potapova, K. M. Varadarajan, A. Richtsfeld, M. Zillich, and M. Vincze, "Attention-driven object detection and segmentation of cluttered table scenes using 2.5 d symmetry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4946–4952.

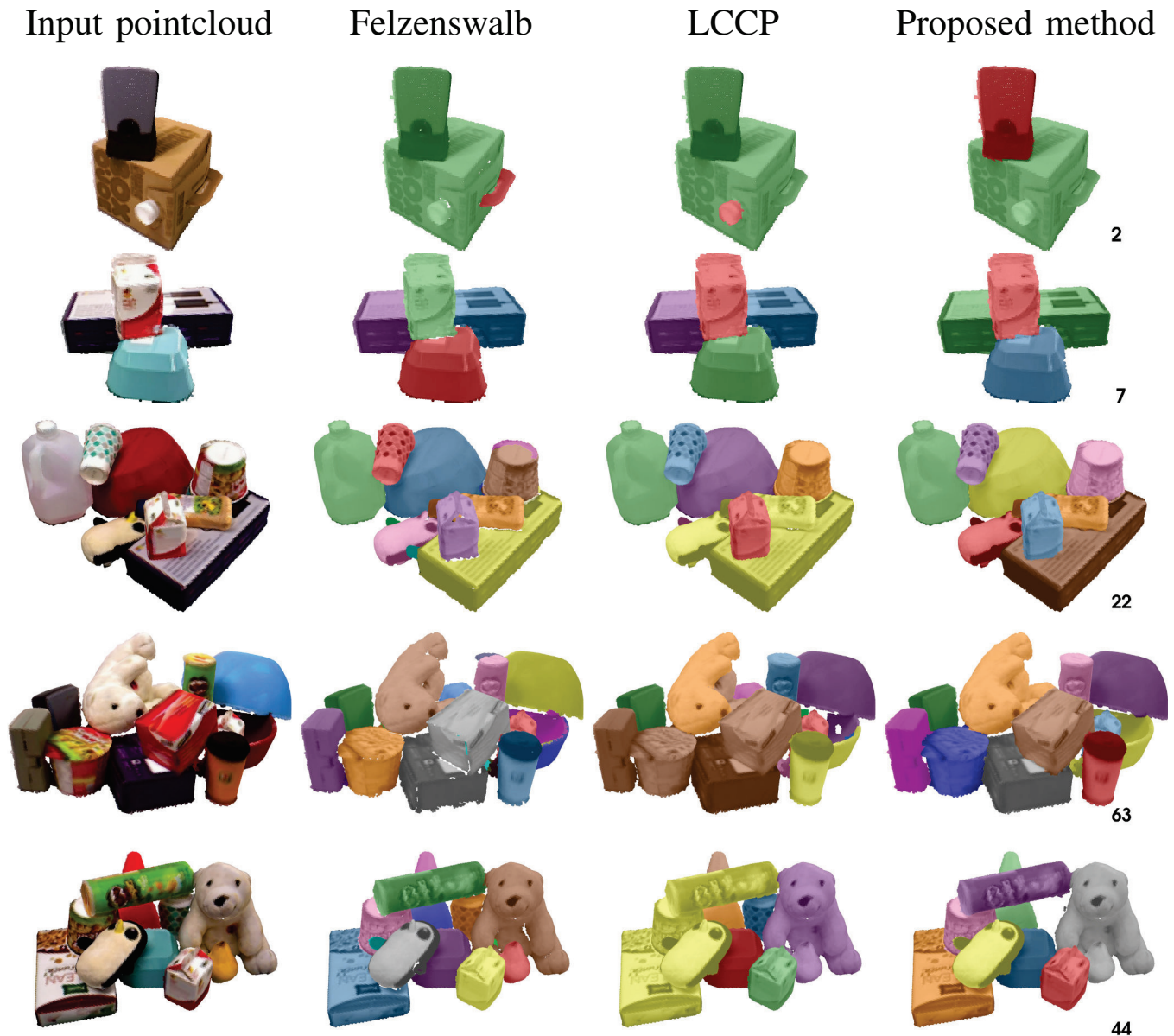| Input pointcloud	Felzenswalb	LCCP	Proposed method

Fig. 8: Comparison of segmentation results on individual scenes. For each method best segments were automatically selected by choosing the predicted segments that have the highest overlap with ground truth masks. For Felzenswalb and LCCP combined segments were used for selection. For our method segments obtained using $s_{occl} = 0.97$ and $s_{smooth} = 0.05$ were used. The numbers show the total number of segments generated by our method for the scene.

[17] C. L. Teo, C. Fermuller, and Y. Aloimonos, "Detection and segmentation of 2d curved reflection symmetric structures," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[18] N. J. Mitra, M. Pauly, M. Wand, and D. Ceylan, "Symmetry in 3d geometry: Extraction and applications," in *EUROGRAPHICS State-of-the-art Report*, 2012. [Online]. Available: http://dx.doi.org/10.1111/cgf.12010

[19] K. Demarsin, D. Vanderstraeten, T. Volodine, and D. Roose, "Detection of closed sharp edges in point clouds using normal estimation and graph theory," *Computer-Aided Design*, vol. 39, no. 4, pp. 276–283, 2007.

[20] 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.

[21] N. J. Mitra, L. J. Guibas, and M. Pauly, "Partial and approximate symmetry detection for 3d geometry," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 560–568, 2006.

[22] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[23] S. Thrun and B. Wegbreit, "Shape from symmetry," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1824–1831.

[24] 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.

[25] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE, 2001, pp. 145–152.