Seeing Behind The Scene: Using Symmetry to Reason About Objects in Cluttered Environments

Aleksandrs Ecins, Cornelia Fermüller and Yiannis Aloimonos

***Abstract*— Symmetry is a common property shared by the majority of man-made objects. This paper presents a novel bottom-up approach for segmenting symmetric objects and recovering their symmetries from 3D pointclouds of natural scenes. Candidate rotational and reflectional symmetries are detected by fitting symmetry axes/planes to the geometry of the smooth surfaces extracted from the scene. Individual symmetries are used as constraints for the foreground segmentation problem that uses symmetry as a global grouping principle. Evaluation on a challenging dataset shows that our approach can reliably segment objects and extract their symmetries from incomplete 3D reconstructions of highly cluttered scenes, outperforming state-of-the-art methods by a wide margin.**

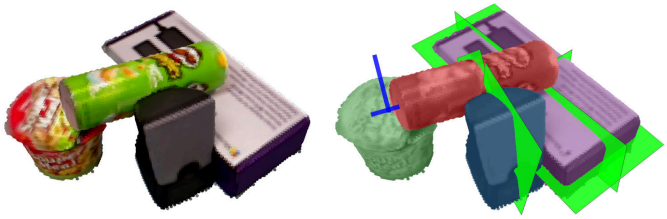


Fig. 1: An output of our segmentation pipeline. Green planes and blue lines denote symmetries corresponding to each segment.

I. INTRODUCTION

Symmetry is ubiquitous in both natural and man-made objects. Gestalt psychologists have recognized the role of symmetry in the human visual system as an important cue that serves as a guide for lower level visual tasks like attention [1] as well as higher level processes such as figure-ground organization [2]. This observation has been mirrored in Computer Vision, where symmetry is used as a preprocessing step for a variety of object-centric perception tasks such as object recognition [3], retrieval [4], reconstruction [5] as well as in robotic applications such as grasping [6]. While most of the existing symmetry detection approaches focus on RGB images, detecting object symmetries from a single 2D image remains an extremely challenging task [7]. This is explained by the fact that even though most of the objects in our environment are symmetric in their 3D shape, the corresponding image patters are rarely symmetric because of the perspective distortion. This issue can be avoided by analyzing 3D shape instead of the 2D images. In addition to simplifying the symmetry detection task, the extracted 3D symmetries can aid further 3D shape analysis of the scene objects.

Building on our previous work [8], we present a novel pipeline for detecting objects and recovering their global symmetries from partial 3D reconstructions of natural scenes. We focus on two symmetry classes that are most commonly encountered in human environments: rotational and reflectional. Symmetry constraints are utilized in two distinct ways, corresponding to the two stages of our algorithm. During the detection stage they act as an attention operator that identifies potential object symmetries in the scene. In the segmentation stage they are used as a grouping mechanism to extract points of the scene that are compatible with the detected symmetries. The two stages work in tandem where symmetry detection is required to initialize the segmentation process which in turn improves the symmetry detection results by filtering out unlikely symmetry candidates.

The input to our system is a pointcloud and a corresponding 3D occupancy map of the scene. Our symmetry detection approach capitalizes on the fact that many objects contain large continuous surfaces which share the same symmetries as the complete object. We oversegment the pointcloud using a region growing algorithm that uses surface normals and point adjacency to enforce a smoothness constraint. Given a segment, its symmetries are detected by aligning symmetry axes/planes to its shape. For rotational symmetry, we define a novel measure of fitness between an oriented point and a symmetry axis that allows fitting a symmetry axis directly to the segment points. Reflectional symmetries are detected using an ICP-like technique that alternates finding correspondences between symmetric points and refining the candidate symmetry plane given the correspondences. Finally, we use a graph-cut approach to segment points of the scene that are consistent with individual detected symmetries. Symmetry is used as a global grouping principle while convexity is used as a local grouping constraint.

We evaluate our complete system on a dataset of heavily cluttered tabletop scenes. We demonstrate that our approach produces precise object segmentation masks and generates accurate symmetry detections in very challenging scenarios. We show that explicitly modeling both rotational and reflectional symmetries leads to significant performance increases over our previous approach that only models reflectional symmetries.

The rest of the paper is organized as follows: Section II gives an overview of current symmetry detection and segmentation methods. Sections III, IV explain our approach to symmetry detection. Section V explains how symmetry

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

(a) (b) (c)

Fig. 2: Pre-segmentation. (a) Scene pointcloud. Pre-segmentation with (b) $\theta_{smooth} = 10^\circ$, and (c) $\theta_{smooth} = 15^\circ$.

is used for object segmentation. Section VI details how reflectional and rotational symmetry stages are combined into a complete pipeline. Finally the evaluation of our approach is presented in Section VII.

II. Related work

A large amount of research in Computer Vision is devoted to symmetry detection in 2D and 3D data. Most of the 2D approaches rely on extracting and matching intensity discontinuities such as edges and point features. One of the first successful methods is the "Generalized Symmetry Transform" of Reisfeld *et al*. [9]. The main idea of the method is that two image points provide evidence for the existence of a symmetry at a midpoint between them if they "reflect" onto each other. This principle was used to generate a continuous "symmetry map" that measures how symmetrical the neighborhood around an image point is. This idea was extended by Loy and Eklundh [10] who employed robust oriented SIFT features to improve the quality of symmetrical matching. Pairs of symmetrically matching feature points were used to generate bilateral and rotational symmetry hypotheses which were then aggregated in a Hough voting scheme to find the dominant symmetries in the image. Some approaches use symmetry to aid object segmentation [11][12]. Teo *et al*. [13] proposed to detect curved bilateral symmetries and add a symmetry term to the Markov Random Field (MRF) used for segmentation.

Most of the early three dimensional methods focused on recovering symmetries from complete 3D models. Podolak *et al*. [4] defined the "Planar Reflective Symmetry Transform" that captured the degree of symmetry of an object with respect to all possible planes passing through it. Exact object symmetries were then detected by extracting the transform maxima. In [14], Mitra *et al*. produced symmetry hypotheses by matching uniquely orientable keypoints on the surface of the mesh and filtered out the dominant ones using mean shift clustering. A number of methods tackled the problem of symmetry detection in partial observations of 3D scenes [15]. Thrun and Wegbeit extracted symmetries for single objects by observing that a valid symmetry should not reflect points of an object into the free space of a scene [16]. Karpathy et al. [17] oversegmented 3D pointcloud reconstructions of office spaces and used segment symmetricity as one of the measures to classify each segment as either object or non-object.

In our previous work [8] we detected reflectional symmetries by matching 3D edge curves and used them to segment objects in 3D pointclouds. In contrast our new approach models both reflectional and rotational symmetries, uses a more elegant and efficient symmetry detection method and proposes a novel strategy for combining results from the two symmetry classes. We make our segmentation code and dataset available on the project website[1].

III. Pre-segmentation

The goal of this stage is to split the scene pointcloud into preliminary segments that will be later used for symmetry detection. Ideally, each segment should belong to the surface of a single object and at the same time be large enough to capture the geometry of the underlying object. To achieve this, we oversegment the input pointcloud into smooth surfaces. Such segments are likely to satisfy the above requirements since object boundaries tend to lie along the surface normal discontinuities.

We employ a variant of the region growing algorithm with a smoothness constraint [18]. Given a seed point six of its nearest neighbors are examined. A neighboring point is considered smooth relative to the seed if the angle between their normals is smaller than a threshold θ_{smooth}. Smooth neighboring points are added to the segment only if at least half of the neighbors satisfy the constraint. This prevents the segments from "leaking" over the surface normal boundaries due to noisy surface normal estimates. The newly added points are used as new seeds. The process continues until no more valid neighbors can be found or all of the pointcloud points belong to a segment. If there are any points remaining they are used to initialize a new segment.

Figure 2 shows segmentation results for two different values of θ_{smooth}, which controls the balance between over and under segmentation. With $\theta_1 = 10^\circ$ the front faces of the red and violet boxes are segmented correctly, while the teddy bear is over-segmented. On the other hand, with $\theta_1 = 15^\circ$ teddy bear is segmented correctly while the box faces are merged. To increase the chances of detecting all of the true object symmetries, we combine the segments from the two

[1] www.umiacs.umd.edu/~aecins/projects/symseg

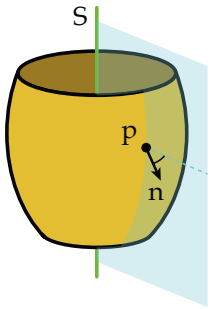


Fig. 3: Constraint imposed on the points of a rotationally symmetric object. The surface normal n of any point of a rotationally symmetric object must lie in the plane formed by the symmetry axis S and the point p itself.

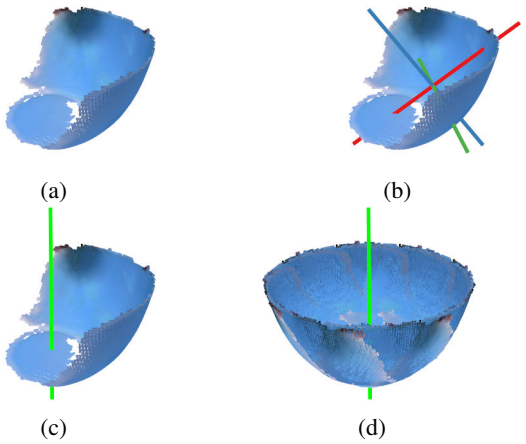


Fig. 4: Rotational symmetry detection stages shown for the blue bowl segment in Figure 2. (a) Input pointcloud. (b) Initial symmetries. (c) Refined symmetry. (d) Cloud reconstructed by rotating the segment around the symmetry axis.

thresholds and feed them to the symmetry detection stage. This does not have a significant impact on the false positive rate of our complete pipeline, since incorrect symmetries can be rejected during the later processing stages.

IV. Symmetry Detection

In order to detect scene symmetries, we employ a geometric fitting approach [19]. Two separate algorithms are used to fit rotational symmetry axes and reflectional symmetry planes respectively to the segments extracted during the pre-segmentation stage. Since the input segments returned by the pre-segmentation stage are noisy, we ensure that both algorithms are robust to outliers and at the same time are sensitive enough to detect symmetries from limited support.

A. Rotational symmetry

Rotational symmetry imposes a strong constraint on the shape of an object. We observe that for any point on the surface of a rotational symmetric object, the corresponding surface normal lies in the plane formed by the symmetry axis and the point itself, as shown on Figure 3. This allows us to formulate an optimization scheme that fits a rotational symmetry axis directly to the segment pointcloud.

Consider a pointcloud consisting of oriented points $P_i = \{p_i, n_i\}$ and a candidate symmetry axis $S^{rot} = \{p^{rot}, d^{rot}\}$ described by a point p^{rot} and a unit direction vector d^{rot}. Let $\angle(S^{rot}, P_i)$ denote the angle between the point normal n_i and the plane formed by the symmetry axis S_{rot} and point p_i. An initial candidate symmetry axis can be aligned to the pointcloud by minimizing the following function over the symmetry axis parameters:

$$S^{rot*} = \arg\min_{S^{rot}} \left(\sum_i \min(\angle(S^{rot}, P_i), \alpha_{max}) \right) \quad (1)$$

where α_{max} is the maximum angle used to limit the influence of outliers. In our experiments we use $\alpha_{max} = 45°$. This function can be minimized using a non-linear solver such as Levenberg-Marquardt.

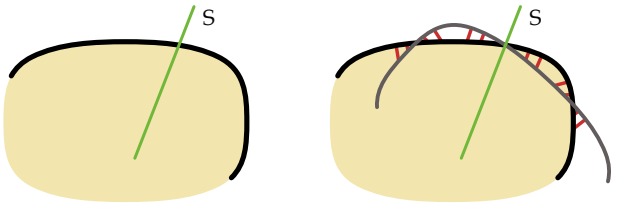
We initialize the process by constructing three candidate symmetry axes that go through the center of mass of the pointcloud and are aligned to its principal axes. After refining all three axes, the one attaining the minimum error of fit is selected as the final hypothesis. This pruning strategy is motivated by the fact that, except for a degenerate case of a sphere, a shape can have at most one rotational symmetry axis. Figure 4 shows the symmetry detected from a partial pointcloud observation of a bowl. To avoid detecting false positives for segments that belong to objects that are not rotationally symmetric (e.g. a box) we reject all symmetries for which the average error of fit is greater than a threshold.

B. Reflectional symmetry

Unlike the case of rotational symmetry, reflectional symmetry does not impose any constraints on the position or surface normal orientation of single points. Instead, relationships are established between pairs of symmetric points. We say that point P'_i is a symmetric correspondence of P_i under the reflectional symmetry S^{refl} if the two points "reflect" onto each other i.e. P'_i is similar to $S^{refl}(P_i)$ - the reflection of P_i by S^{refl}. Inspired by Iterative Closest Point algorithm (ICP) [20], our fitting approach alternates between symmetric correspondence estimation and symmetry plane refinement, as shown in Figure 5.

Let $S^{refl} = \{p^{refl}, n^{refl}\}$ denote a symmetry plane represented by a point p^{refl} lying in the plane and the plane's normal n^{refl}. Given a point in a segment, we find its symmetric correspondence by reflecting it across the symmetry plane and searching for the nearest neighbor within a fixed radius (1cm in our experiments). Note that we are not guaranteed to find a correspondence for every point in the segment due to the occlusions in the scene. Once the symmetric correspondences are established, we refine the symmetry plane by minimizing the point-to-plane distance between the reflected points and its correspondences:

$$S^{refl*} = \arg\min_{S^{refl}} \left(\sum_{\{i\}} d_{p-pl}(S^{refl}(P_i), P'_i) \right) \quad (2)$$



(a) (b)

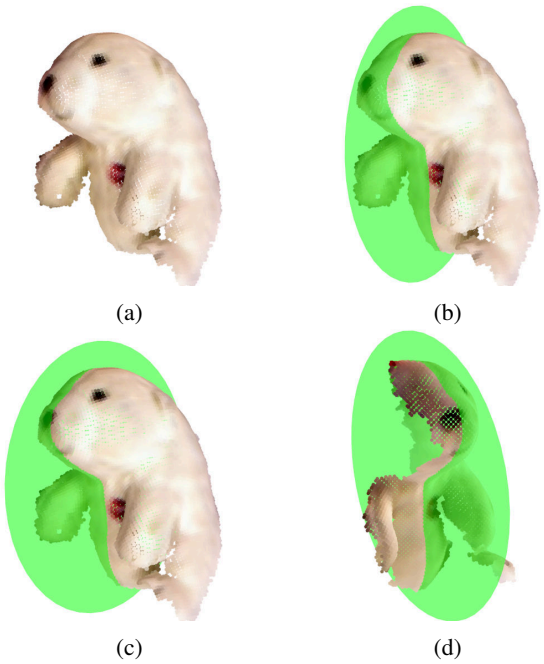
Fig. 5: A candidate symmetry plane is refined by aligning a segment to its own reflection. (a) Input object pointcloud and candidate symmetry. Black curve denotes the observed points of the object. (b) Grey curve denotes the reflected pointcloud and red lines show the estimated symmetric correspondences.

where d_{p-pl} stands for point-to-plane distance and $\{i\}$ is the set of points that have a symmetric correspondence given symmetry candidate S^{refl}. The process is repeated until convergence, or until a maximum of 20 iterations is reached. Provided a good initial symmetry guess, this fitting approach results in very accurate alignment of a symmetry plane to a segment. To bootstrap the refinement process we generate a set of initial symmetry planes that pass through the center of mass of the segment. Plane normal vectors are sampled uniformly from a unit sphere in steps of 36° in azimuth and elevation. This results in a set of 25 initial symmetry planes. We then apply the iterative refinement approach described above to the individual candidates. Unlike the case of rotational symmetry we do not select a single "best" symmetry and instead return all refined symmetries with an error of fit lower than a threshold. This reflects the fact that a single object can have multiple reflectional symmetries. Figure 6 shows the symmetry detected for a teddy bear.

V. Segmentation

Given a single symmetry hypothesis, the goal of the segmentation step is to extract scene points that belong to the object possessing the symmetry (if the detected symmetry is correct), or to decide that a symmetry is not valid (if the detected symmetry is a false positive). We use two principles to judge whether a single point is compatible with either a rotational or a reflectional symmetry. Firstly, a point must satisfy the rotational/reflectional symmetry fitness measure that was used for symmetry detection. Secondly, a point rotated/reflected by the symmetry must lie in the occluded space of the scene [16]. To capture these properties, we define two pointwise measures: symmetry score $Sym(P)$ and occlusion score $Ocl(P)$. These scores are calculated differently for rotational and reflectional symmetry, but in both cases, the scores are normalized to lie in the $[0,1]$ range and a lower score indicates better compatibility with a symmetry hypothesis.

To segment an object we use a graph-cuts approach that combines the global grouping principles of symmetry fitness and occlusion with the local grouping principle of convexity [21]. A graph is defined over the points in the input pointcloud. Given a single symmetry hypothesis, each point



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

Fig. 6: Reflectional symmetry detection stages shown for a teddy bear segment in Figure 2. (a) Input pointcloud. (b) One of the 25 initial symmetries. (c) Refined initial symmetry. (d) Side view of the refined initial symmetry.

in the scene is labeled as foreground (fg) or background (bg) by minimizing the following energy functional:

$$E(f) = \sum_{p \in \mathscr{P}} D_p(f_p) + \lambda \sum_{\{p_1,p_2\} \in \mathscr{N}} V_{p_1,p_2} \cdot \delta(f_{p_1} \neq f_{p_2}) \tag{3}$$

The binary term V_{p_1,p_2} ensures that segment boundaries are more likely to occur at concave discontinuities in the pointcloud. It does not depend on the symmetry type, and is computed in the same way for rotational and reflectional symmetries. We establish a point adjacency graph $\mathscr{N}$ by connecting every point to 9 of its nearest neighbors. Surface normals are then used to compute the weights between adjacent points:

$$V_{p_1,p_2} = \begin{cases} \exp\left(\dfrac{n_1 \cdot n_2}{\sigma_{convex}}\right) & \text{if } n_1 \cdot (p_1 - p_2) > 0 \\ \exp\left(\dfrac{n_1 \cdot n_2}{\sigma_{concave}}\right) & \text{otherwise} \end{cases} \tag{4}$$

where $\sigma_{convex} = 2$ and $\sigma_{concave} = 0.15$.

The role of the unary term $D_p(f_p)$ is to enforce the symmetry constraints. We now explain how it is calculated for rotational and reflectional symmetries respectively.

A. Rotational symmetry

As discussed in section IV-A, the fitness score between an oriented point P and a rotational symmetry S^{rot} is calculated as the normalized angle between the point normal and the plane containing the symmetry axis and the point itself:

$$Sym^{rot}(P) = \frac{\min(\angle(S^{rot}, P), \alpha_{max})}{\alpha_{max}} \tag{5}$$

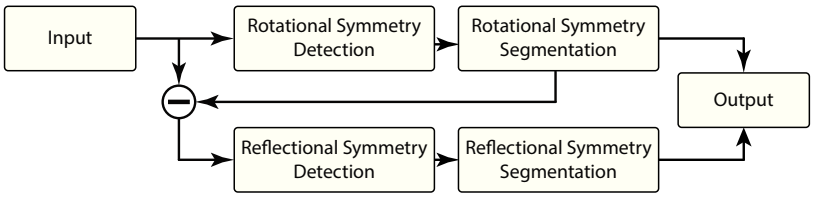


Fig. 7: System flowchart. Note that the rotationally symmetric segments are subtracted from the pointcloud before the reflectional symmetry stage.

For the occlusion score, we want to penalize points that "stick out" into free space after being rotated around the symmetry axis. Let $S^{rot}(P,\theta)$ denote a point constructed by rotating P around S^{rot} by angle θ and let $EDT(S^{rot}(P,\theta))$ denote the distance from that point to the nearest occluded voxel. Occlusion score is calculated by repeatedly rotating P and taking the maximum distance:

$$Ocl^{rot}(P) = \frac{\max_i(EDT(S^{rot}(P,\theta * i)))}{d_{max}}, \tag{6}$$

where $\theta = 30^\circ$ and $n \in \{1,...,12\}$. Nearest occluded voxel lookup can be done efficiently by constructing a three dimensional Euclidean Distance Transform of the occupancy map of the scene [22]. $d_{max} = 3cm$ is a clamping factor used for the EDT.

The unary weights for a rotational symmetry are set as:

$$D_p(fg) = (1 - Sym^{rot}(P)) \tag{7}$$
$$D_p(bg) = Ocl^{rot}(P) \tag{8}$$

These terms encode the intuition that a point belonging to the surface of a rotationally symmetric object has to satisfy both the symmetry fitness and the occlusion constraints.

B. Reflectional symmetry

In the case of reflectional symmetry, we can only measure the fitness of a point P to a symmetry S^{refl} if it has a symmetric correspondence P'. Fitness score is calculated as the normalized angle between the reflected point normal and the corresponding point normal:

$$Sym^{refl}(P) = \frac{\min(\angle(S^{refl}(n), n'), \alpha_{max})}{\alpha_{max}} \tag{9}$$

where $\alpha_{max} = 60^\circ$ reduces the influence of outliers.

The occlusion score, which is defined for all of the points in the scene, is computed by reflecting the points by the symmetry plane and checking the distance to the nearest occluded voxel:

$$Ocl^{refl}(P) = \frac{EDT(P')}{d_{max}} \tag{10}$$

For a point that has a symmetric neighbor, the unary weights are computed similarly to the rotational symmetry case (equations 7 and 8). Points that have no symmetric correspondence are treated differently. Since the symmetry fitness score is unavailable, the foreground weight is set to 0, while the occlusion score is used for the background weight:

$$D_p(fg) = 0 \tag{11}$$
$$D_p(bg) = Ocl^{refl}(P) \tag{12}$$

C. Hypothesis filtering

For a given symmetry, the output of our segmentation stage is a segmentation mask H associated with the symmetry hypothesis S used to generate it. To reduce the number of false positive segments returned, we evaluate each segment on how well it adheres to the grouping principles used in the segmentation process. Three measures are used:

- **Symmetry score**. Object segments must be symmetric. The score is calculated as the average symmetry score for all of the points of the segment:

$$Sym(H) = \frac{1}{|H|}\sum_{P_i \in H} Sym(P_i) \tag{13}$$

- **Occlusion score**. All points of the object segment must rotate/reflect into occluded space. This is measured as the average of the pointwise occlusion scores:

$$Ocl(H) = \frac{1}{|H|}\sum_{P_i \in H} Ocl(P_i) \tag{14}$$

- **Smoothness score.** Object segment boundaries tend to coincide with depth discontinuities, and are unlikely to lie on flat surfaces. Smoothness score is calculated as the sum of the weights between adjacent points that were severed by the segment, normalized by the segment size:

$$Smooth(H) = \frac{1}{|H|}\sum_{\{P_i,P_j\} \in \mathcal{N}_H} V_{P_i,P_j} \tag{15}$$

A valid object segment must satisfy all three measures: $Sym(H) < s_{sym}$, $Ocl(H) < s_{occl}$ and $Smooth(H) < s_{smooth}$. Segments that fail any of the tests are discarded. Changing the filtering thresholds allows us to tune our system to the desired precision-recall trade-off.

VI. Pipeline

In this section we present our complete system pipeline that combines the symmetry detection and segmentation stages for both symmetry types. Figure 7 shows the flowchart of the system. It consists of two sequential streams: one for rotational and the other for reflectional symmetries. Given an input pointcloud and an occupancy map of the scene,

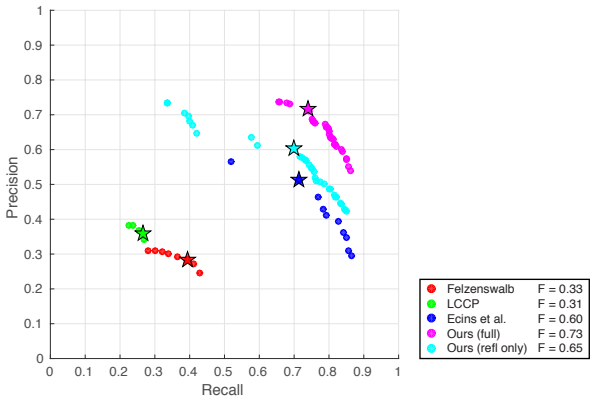


Fig. 9: Segmentation evaluation. Stars mark the points of maximum F-measure.

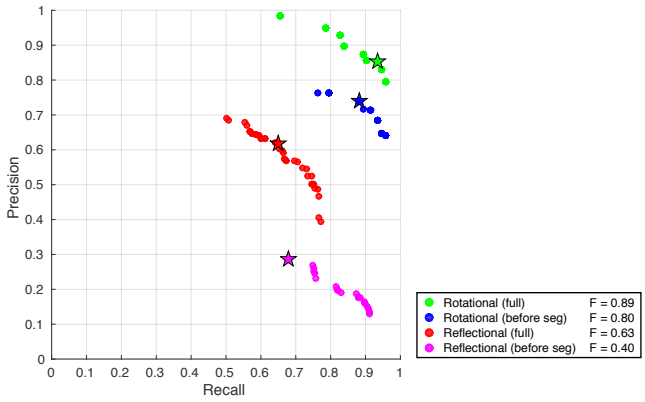


Fig. 10: Symmetry detection evaluation. Stars mark the points of maximum F-measure.

rotational objects are detected first. Next the points belonging to the detected rotational object segments are removed from the input pointcloud, before passing it to the reflectional symmetry stage. This is done to avoid detecting reflectional symmetries on rotationally symmetric surfaces, since a rotationally symmetric object has an infinite number of valid reflectional symmetries that pass through its symmetry axis. After segmenting both rotational and reflectional objects we combine the outputs from both stages. Note that the segments returned by our approach may be overlapping. For objects having multiple symmetries (e.g. a box) our system should return three identical segments corresponding to different symmetries of the same object. To handle these cases, we merge the segments for which the intersection over union is greater than 95% and concatenate the symmetries used to generate them. This allows our pipeline to return segments associated with multiple symmetries.

VII. Experiments and Evaluation

We evaluate our approach both in terms of its ability to segment objects as well as its ability to detect object symmetries. We use the Cluttered Tabletop Dataset [8] for evaluation. This dataset consists of 89 reconstructions of tabletop scenes collected from multiple views using a Kinect sensor. Due to the high amount of clutter and limited viewing angles, objects are only partially visible and are always physically adjacent to other objects, which makes this dataset especially challenging for our task. The dataset contains ground truth segmentation masks for the scene objects. To test the symmetry detection performance of our approach, we annotated the dataset with per-object rotational and reflectional symmetries.

A. Segmentation

An object is considered to be detected correctly if the intersection over the union score between its ground truth segmentation mask and a predicted mask is greater than 90%. Recall is measured as the fraction of objects in the dataset that were segmented correctly, while precision is measured as the ratio of correctly predicted objects to the total number of predictions. We evaluate two versions of our approach. The first version is our complete pipeline *Ours (full)*. To investigate the importance of modeling rotational symmetries, we also test a reduced version of our pipeline that only uses reflectional symmetry to segment the scenes *Ours (refl only)*.

We compare our approach to 3 other algorithms. The first two algorithms are the *LCCP* [21] and an adaptation of the *Felzenswalb* segmentation algorithm [23] to pointcloud data. Similarly to our method, both approaches are completely bottom-up, but unlike ours, they rely solely on local convexity principle to segment the objects. The third algorithm is our previous work [8] that uses reflectional symmetry as the grouping principle.

The PR curves for the evaluated algorithms are generated by sweeping the algorithm parameters and computing the convex hull over the resulting precision/recall point set (Figure 9). *LCCP* and *Felzenswalb* achieve a similar performance with maximum F-measures of 31% and 33% respectively. Both methods are significantly outperformed by the symmetry-based methods. This shows that local convexity alone is not sufficient for accurate object segmentation in situations where objects are in close physical contact, and are not restricted to be concave. On the other hand, symmetry provides a global object-level grouping principle, which allows symmetry-based methods to better handle such complicated scenarios.

All three symmetry-based methods have very similar recall characteristics, but differ in their precision. Our complete pipeline *Ours (full)* achieves the highest F-score of 73%. Methods relying solely on the reflectional symmetry, *Ours(refl only)* and *Ecins et al.*, achieve similar recall, but have a lower precision. This can be attributed to two reasons. First, approaches that do not explicitly model rotational symmetry can only segment rotational objects if they happen to detect a reflectional symmetry that goes through the rotational symmetry axis. Even if such symmetry is detected, reflectional segmentation imposes fewer constraints on the segment, compared to a rotational method. As a result it

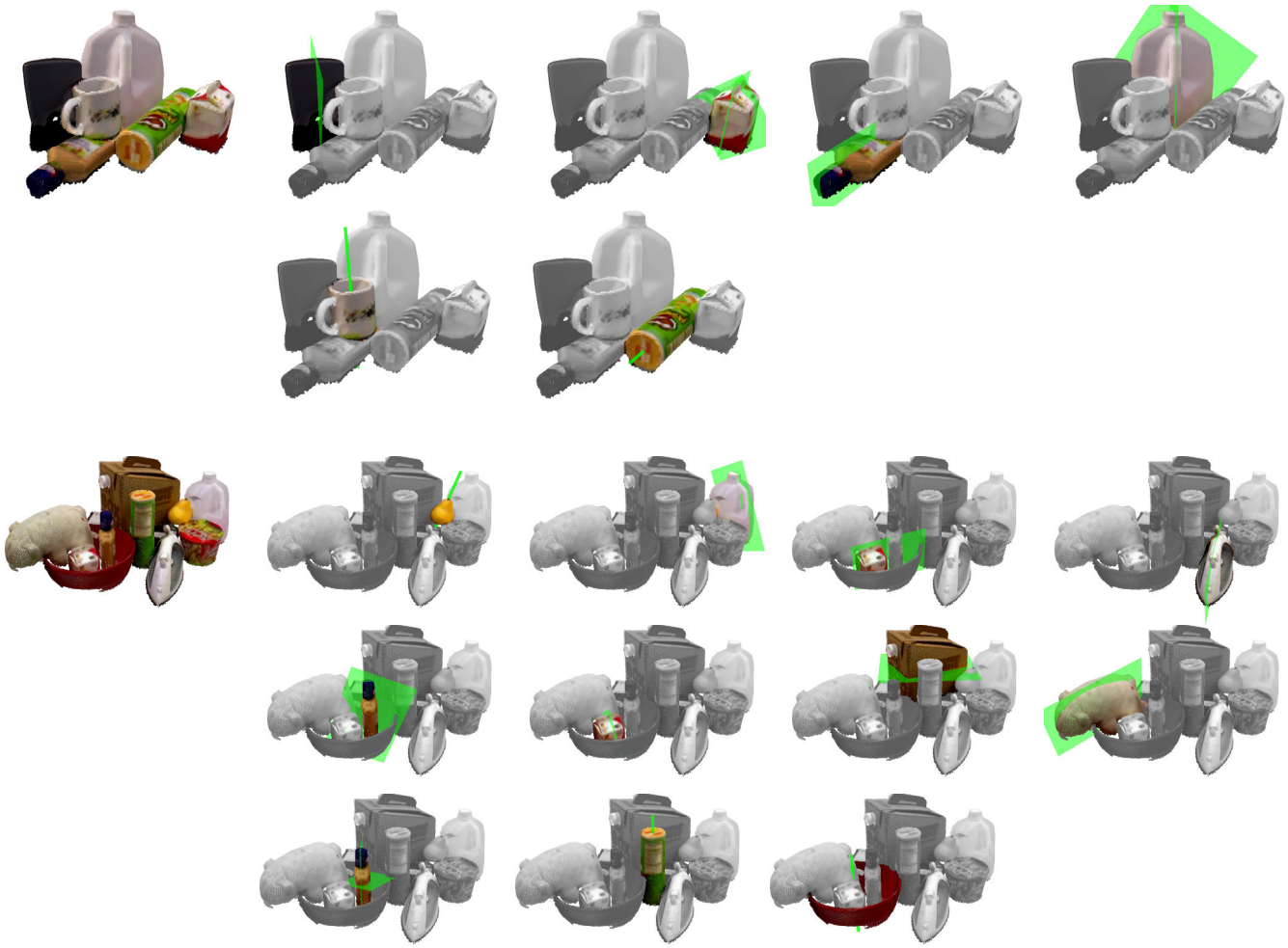


Fig. 8: Sample detection results of our pipeline. First image in a row shows the input pointcloud. Following images show the detected objects masks and their symmetries.

is less likely to segment it correctly. Second, additional segmentation errors arise if the axis of a rotationally symmetric objects lies in the symmetry plane of a different objects. The two will be merged together by a reflectional segmentation approach since both objects "share" the same reflectional symmetry. *Ours (refl only)* achieves an F-score of 65% outperforming *Ecins et al.* with 60%. The difference in performance can be explained by the fact that our surface-based reflectional symmetry detection method has a lower false-positive rate compared to a more noisy curve-based method used by *Ecins et al.*.

B. Symmetry detection

Our symmetry detection evaluation procedure is inspired by the evaluation method used for 2D symmetry detection [24]. Similarity between a predicted and ground truth symmetries is measured based on their angular and positional differences. For rotational symmetry, we calculate the angle between the two symmetry axes, and the distance between the projections of the ground truth segment centroid on the two axes. In the case of reflectional symmetry, we use the angle between the planes and the distance between projections of the ground truth segment onto the two planes. For both symmetry types, a detected symmetry is considered correct if the the angular difference is less than 10° and the positional difference is less than 1 cm.

We investigate the symmetry detection performance of our complete pipeline. For each of the scenes, symmetries associated with the returned segments are concatenated together and treated as outputs of a symmetry detection stage. This is compared to the detection results provided directly by the symmetry detection stages, without the following segmentation step.

Figure 10 shows the PR curves for symmetry detection evaluation. Rotational symmetries are detected very reliably with an F-score of 89%, while reflectional symmetries get a more modest score of 63%. For both symmetry types, our pipeline significantly outperforms the raw detections from the symmetry detection stages. This demonstrates, that segmentation can be viewed as a reliable filtering stage for the detected symmetries. The performance gap between pre and post segmentation results is particularly notable for reflectional symmetries - an increase of 23%. This is due to the fact that our pipeline removes rotationally symmetric

parts of the scene before detecting reflectional symmetries. This has an effect of "decluttering" the scene, making further processing easier.

C. Runtime analysis

We evaluated the runtime of a C++ implementation of our algorithm on an AMD 1700x CPU. Table I shows average runtimes on scenes of different sizes from the CTD dataset. Scenes with 1-4 objects take less than 2 seconds to process, while larger scenes with 9-12 objects take 5.46 seconds. In general the runtime of our algorithm increases with the number of objects in the scene, since the number of segments produced by the pre-segmentation step is greater for larger and more geometrically complex scenes.

Pipeline stage	1-4 objects	5-8 objects	9-12 objects
Rotational stage	0.20 sec	0.53 sec	0.80 sec
Reflectional stage	1.70 sec	3.53 sec	4.66 sec
Total	1.90 sec	4.06 sec	5.46 sec

TABLE I: Average runtimes for different stages of our pipeline on the scenes from the Cluttered Tabletop Dataset.

VIII. Conclusions

This paper presents a holistic approach for segmenting rotational and reflectional symmetric objects in 3D point-clouds of natural scenes. Combining a novel symmetry fitting method with a robust segmentation stage, our approach can reliably segment and extract object symmetries in cluttered scenes, where complete object shape information is not available. Our approach is bottom-up, requiring no prior knowledge about the objects. The performance of the algorithm can be improved even further by incorporating color cues into the segmentation process. Another avenue for improvement is to model additional "combinatorial" symmetry classes that correspond to a more limited, yet common set of object classes. An example is a set of 3 perpendicular planes which correspond to box-like objects. An obvious development of our approach is to use the detected symmetry axes/planes to reconstruct the occluded parts of the scene objects. This has direct use-cases in multiple fields from robotics, where complete shape information aids grasp planning [6], to augmented reality, where virtual objects need to interact realistically with the objects in the user's surroundings.

IX. Acknowledgements

The authors would like to Konstantinos Zampogiannis for helping benchmark the runtime of the algorithm. The support of the National Science Foundation under awards SMA 1540916 and CNS 1544787 and the support of ONR under award N00014-17-1-2622 are greatly acknowledged.

References

[1] G. Kootstra and L. R. Schomaker, "Prediction of human eye fixations using symmetry," in *The 31st Annual Conference of the Cognitive Science Society (CogSci09)*. Cognitive Science Society, 2009, pp. 56–61.

[2] 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.

[3] E. Barnea and O. Ben-Shahar, "Depth based object detection from partial pose estimation of symmetric objects," in *European Conference on Computer Vision*. Springer, 2014, pp. 377–390.

[4] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser, "A planar-reflective symmetry transform for 3d shapes," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 549–559, 2006.

[5] P. Speciale, M. R. Oswald, A. Cohen, and M. Pollefeys, "A symmetry prior for convex variational 3d reconstruction," in *European Conference on Computer Vision*. Springer, 2016, pp. 313–328.

[6] A. H. Quispe, B. Milville, M. A. Gutiérrez, C. Erdogan, M. Stilman, H. Christensen, and H. B. Amor, "Exploiting symmetries and extrusions for grasping household objects," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3702–3708.

[7] C. Funk, S. Lee, M. R. Oswald, S. Tsogkas, W. Shen, A. Cohen, S. Dickinson, Y. Liu, *et al.*, "2017 iccv challenge: Detecting symmetry in the wild."

[8] A. Ecins, C. Fermüller, and Y. Aloimonos, "Cluttered scene segmentation using the symmetry constraint," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2271–2278.

[9] D. Reisfeld, H. Wolfson, and Y. Yeshurun, "Context-free attentional operators: the generalized symmetry transform," *International Journal of Computer Vision*, vol. 14, no. 2, pp. 119–130, 1995.

[10] G. Loy and J.-O. Eklundh, "Detecting symmetry and symmetric constellations of features," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 508–521.

[11] H. Fu, X. Cao, Z. Tu, and D. Lin, "Symmetry constraint for foreground extraction," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 44, no. 5, 2014.

[12] T. Riklin-Raviv, N. Kiryati, and N. Sochen, "Segmentation by level sets and symmetry," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 1015–1022.

[13] C. L. Teo, C. Fermuller, and Y. Aloimonos, "Detection and segmentation of 2d curved reflection symmetric structures," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1644–1652.

[14] 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.

[15] L. Yang, H. Uchiyama, J. M. Normand, G. Moreau, H. Nagahara, and R. i. Taniguchi, "Real-time surface of revolution reconstruction on dense slam," in *2016 Fourth International Conference on 3D Vision (3DV)*, Oct 2016, pp. 28–36.

[16] 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.

[17] 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.

[18] F. v. d. H. Rabbani, Tahir and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," pp. 248–253, 2006.

[19] A. Ecins, C. Fermller, and Y. Aloimonos, "Detecting reflectional symmetries in 3d data through symmetrical fitting," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, Oct 2017, pp. 1779–1783.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] J. Liu, G. Slota, G. Zheng, Z. Wu, M. Park, S. Lee, I. Rauschert, and Y. Liu, "Symmetry detection from realworld images competition 2013: Summary and results," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2013.