SpelsNet: Surface Primitive Elements Segmentation by B-Rep Graph Structure Supervision Supplementary Material

Anonymous Author(s)
Affiliation
Address
email

1   Metrics

The performance of SpelsNetis evaluated with mean IoU $sIoU$ over matched segments with and their types $tIoU$ in accordance with previous works Huang et al. [2021], Yan et al. [2021], Sharma et al. [2020]. Similarly, given the predicted elements, we compare them with the ground truth data by first building a matching and its segmentation membership $\mathbf{W}_f$ between corresponding elements and then defining a metric with respect to this matching. The correspondence between the ground truth elements and the predicted elements is constructed by bipartite matching with a relaxed IoU cost over ground truth and predicted labels following Sharma et al. [2020].

$$sIoU = \frac{1}{K} \sum_{k=1}^{K} IoU(\tilde{\mathbf{W}}_{:,k}, \mathbf{W}_{:,k})$$

$$tIoU = \frac{1}{K} \sum_{k=1}^{K} \mathcal{I}[\tilde{\mathbf{T}}_k = \mathbf{T}_k],$$

where K is the number of matched segments, $\mathcal{I}$ is the indicator function, $\mathbf{T}_k$ and $\tilde{\mathbf{T}}_k$ are the GT and predicted k_{th} segment types.

2   Comparisons

As the data for PrimitiveNet was shared partly, we have retrained PrimitiveNet on exactly the same ABCParts-VEF data and splits as in our experiments with SpelsNet. The default settings described in the paper were used during training. This allows us to compare the same metrics on exactly the same data for both methods. The type labels of primitives include both curves and patches, thus are able to present the metrics for curve types of primitive (lines, circles, ellipses, splines). PrimitiveNet, originally, was not designed to predict curve types, but rather the boundary points.

The qualitative results for ComplexGen presented in Figure 4 of the paper were obtained from the assets publicly shared by the authors. The visualization process directly follows the instructions shared in their implementation.

3   Datasets Specifics

The experiments presented in the main paper, are conducted on two datasets. The Table 1 summarizes the average number of curve and patch elements per model. The sparsity is the average ration of non-zero elements to a total number of elements in B-RepLAR^{pcd} matrices. The ratio of total number of elements by their type is given in Table 2. Compared to ABCParts, the CC3D dataset has no

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

ellipses as curves. The percentage of unknown types (those that can not be assigned to the basic ones) is also zero for CC3D.

| Dataset | Number of edges per model | | Number of faces per model | | sparsity | |
	mean	max	mean	max	M_1	M_2
ABCParts-VEF	43.51 ± 43.85	4467	8.24 ± 16.62	1757	0.153 ± 0.157	0.248 ± 0.182
CC3D-VEF	368.96 ± 874.283	35856	143.10 ± 339.96	17176	0.061 ± 0.091	0.132 ± 0.163

Table 1: The average number of B-Rep elements per model in two datasets and the average sparsity of LAR^{pcd}.

| Dataset | Ratio of edge type (%) | | | | | Ratio of face type (%) | | | | | | |
	Line	Circle	Spline	Ellipse	Unknown	Plane	Cylinder	Spline	Sphere	Cone	Torus	Unknown
ABCParts-VEF	38.31	26.91	31.12	2.36	1.31	42.99	25.57	17.18	1.59	4.45	5.09	3.14
CC3D-VEF	56.58	27.20	16.22	0	0	48.57	27.63	11.76	2.27	4.08	5.70	0

Table 2: Primitive types distribution in two datasets.

3.1   ABCParts-VEF Preparation

Originally, the ABCParts dataset offers the labeling of surfaces patches only, where the surface types are contained in 10 classes (with cases of extrusion, revolution surfaces). During the evaluation this set is reduced to the basic set which includes plane, cylinder, sphere, torus, and the rest classes are considered as spline surfaces. We prepare the updated version of this dataset ABCParts-VEF with corrected patch types, adding curve types and extending it with B-Rep structural information in the form of characteristic matrices $\mathbf{M}_1^p$ and $\mathbf{M}_2^p$. The parametrization of curves and surface patches gives the geometric details of each element. The data goes through labeling with fixing some common errors as splitting multiples bodies correctly and removing the construction curves specific to Open CASCADE Laughlin [2020](*e.g.*, the extra edge of a cylinder). The required B-Rep incidence relations in forms of characteristic matrices are extracted directly from STEP formats of the data models. The labeling information, including the topological relations of B-Rep elements, is transferred into mesh representation of this dataset in nearest neighbour manner per triangle under the tolerance threshold τ. In Table 2 we summarize the percentage of different element types for two datasets ABCParts and CC3D for surface patches among Plane, Cylinder, Sphere, BSpline Surface, Cone, Torus and Unknown, and curves among Line, Circle, BSpline, Ellipse and Unknown. The distribution of those elements is highly unbalanced.

3.2   Scanned Data Challenges

The Figure 1 illustrates the possible scanning artifacts that may present in 3D scanned CAD counterpart. Among them are the missing parts, protrusions, edge smoothing and surface noise. This makes this dataset more challenging, and oriented towards real Scan-to-Brep scenario.

3.3   Results on Real Scanned Data

We present the visual results of segmentation of our SpelsNet and PrimitiveNet methods on several real scans. The scans are downloaded from a 3D content sharing platform. These data has no ground truth labeling. Both methods were pretrained on ABCParts dataset. We can observe that the types are mostly predicted as spline surfaces, which tells us that the local characteristics of points distribution are significantly different for synthetic and scanned data. The over- and under-segmentation illustrates well the specifics of the real scanned data such as the surface noise and smoothness of edges. The results of our method look more concise and topologically justified, specially in the edges prediction, as the topology of B-Repelements is explicitly supervised in our method. The training or finetuning on CC3D dataset should help to significantly improve the performance of our method on real scans.

2
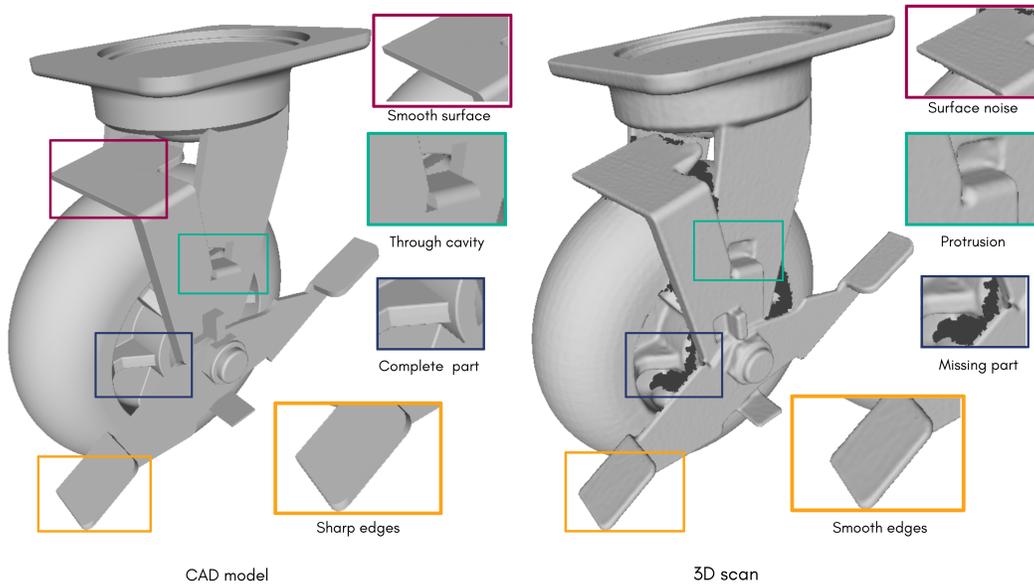
Figure 1: CC3D scanning artifacts: missing parts, protrusions, edge smoothing, surface noise artifacts. The left is a CAD triangulated model, on the right is its 3D scan.
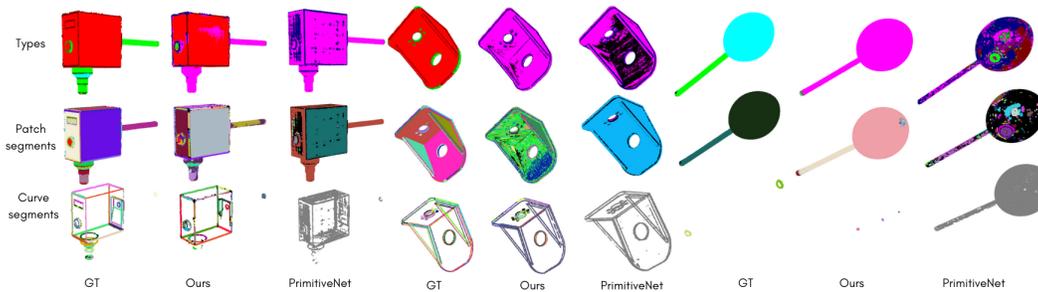
Figure 2: Visual results of comparison our method and PrimitiveNet on a test set of CC3D dataset.
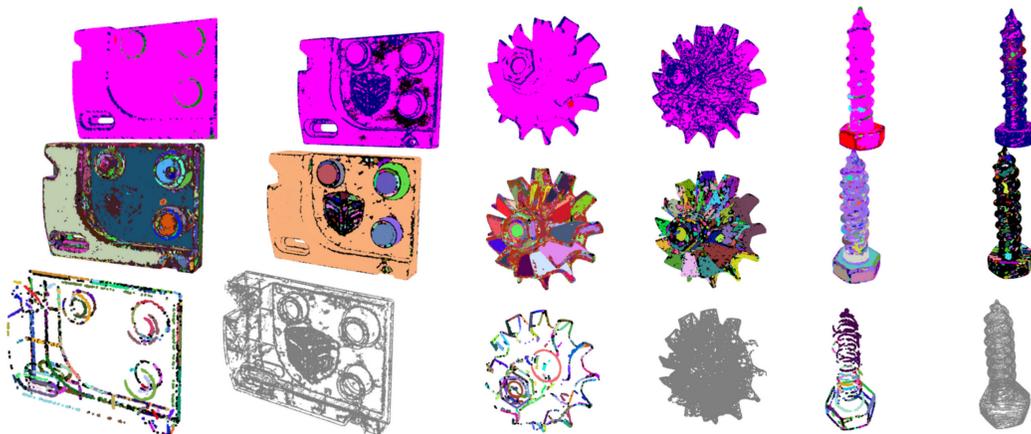
Figure 3: More results on real scanned data. The top row shows the predicted types, the middle row is the surface patches segmentation, the bottom is the edge segments(in our case) and the boundary points for PrimitiveNet. Left column in each pair is our SpelsNet, on the right are the results of PrimitiveNet.

3

References

J. Huang, Y. Zhang, and M. Sun. Primitivenet: Primitive instance segmentation with local primitive embedding under adversarial metric. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15343–15353, 2021.

T. Laughlin. pyocct – python bindings for opencascade via pybind11, 2020. https://github.com/trelau/pyOCCT.

G. Sharma, D. Liu, S. Maji, E. Kalogerakis, S. Chaudhuri, and R. Měch. Parsenet: A parametric surface fitting network for 3d point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 261–276. Springer, 2020.

S. Yan, Z. Yang, C. Ma, H. Huang, E. Vouga, and Q. Huang. Hpnet: Deep primitive segmentation using hybrid representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2753–2762, 2021.

4