

Noise-resistant Unsupervised Object Segmentation in Multi-view Indoor Point Clouds - Supplementary Material

Dmytro Bobkov¹, Sili Chen², Martin Kiechle³, Sebastian Hilsenbeck⁴ and Eckehard Steinbach¹

¹*Chair of Media Technology, Technical University of Munich, Arcisstr. 21, Munich, Germany*

²*Institute of Deep Learning, Baidu Inc., Xibeiwang East Road, 10, Beijing, China*

³*Chair of Data Processing, Technical University of Munich, Arcisstr. 21, Munich, Germany*

⁴*NavVis GmbH, Blutenburgstr. 18, Munich, Germany*

{dmytro.bobkov, martin.kiechle, eckehard.steinbach}@tum.de, sili.chen90@gmail.com, sebastian.hilsenbeck@navvis.com

Keywords: Object Segmentation, Concavity Criterion, Laser Scanner, Point Cloud, Segmentation Dataset.

In this document we first give additional details on how to generate appropriate ground truth data for the proposed multi-scale evaluation approach and evaluation metric. In the second part we provide additional discussion of the chosen parameter values for the concavity criterion, which is omitted from the main part due to lack of space. After that, we provide experimental results on further three indoor scenes of the proposed pointcloud dataset. Finally, we also provide additional results on the dataset of Lai *et al.*

1 Metric for Ground Truth Labelling and Evaluation

As described in the paper, we designed a new multi-scale ground truth (GT) evaluation approach. We label each of meaningful object parts as a separate semantic object within fine GT. In order to generate coarse GT, we merge appropriate object parts into one label.

The approach to establish correspondences between points and appropriate GT labels is presented in Algorithm 1. Here, we have several data structures as input arguments of the algorithm. The first one IL describes mapping from points to fine GT labels. The second one FC specifies mapping from fine to coarse GT labels. The third one FP describes correspondences from fine GT labels to predicted labels, where a predicted label corresponding to a fine label is the majority of predicted labels of this part. We however are interested in establishing correspondences between points and appropriate fine GT (possibly merged) labels, which are returned in map IM .

We show a simple example of input data for the algorithm in Fig. 1a. Here 10 points need to be evaluated according to the given predicted labels, coarse and fine GT data. FM and IM are empty prior to algorithm execution. After the algorithm has finished, the FC and FP have not changed (see Fig. 1b). Furthermore, the returned structure IM describes mapping from point indices to merged fine labels.

Now, when the proper GT data has been generated, we can directly extend the metric of Richtsfeld *et al.* (?) to multi-scale objects. The multi-scale over- and undersegmentation errors can be calculated as follows:

$$ME_{os} = 1 - \frac{\sum_{i=1}^n PT_i}{\sum_{i=1}^n M_i}, \quad (1)$$

$$ME_{us} = \frac{\sum_{i=1}^n (P_i - PT_i)}{\sum_{i=1}^n M_i}, \quad (2)$$

where M_i is the number of points with the i th merged fine label, P_i is the number of points with the predicted label that after the mapping from merged fine label to predicted label is corresponding to the i th merged fine label. Furthermore, PT_i is the number of correctly assigned point labels within the i th merged fine label. Finally, n is the number of distinct merged fine labels.

For the example given in Fig. 1 we compute the above defined errors as follows:

$$ME_{os} = 1 - \frac{5+1+2+2}{5+1+2+2} = 0, \quad (3)$$

$$ME_{us} = \frac{(6-5)+(6-1)+(2-2)+(2-2)}{5+1+2+2} = 0.6. \quad (4)$$

Algorithm 1 Generate Appropriate GT Labels

Input arguments:

Map $IL < PointId, FineL >$

Map $FC < FineL, CoarseL >$

Map $FP < FineL, PredL >$

Procedure:

$keyArray \leftarrow keys(FP)$

$K \leftarrow keyArray.size$

Map $FM < FineL, MergedFineL > \leftarrow \{ \}$

Map $IM < PointId, MergedFineL > \leftarrow \{ \}$

$flagArray \leftarrow new_array(K)$

$flagArray \leftarrow 0$

for $k = 0$ to $K - 1$ **do**

if $flagArray[k] == 0$ **then**

for $j = k + 1$ to $K - 1$ **do**

$RFlag \leftarrow (FP[keyArray(k)] == FP[keyArray(j)]) \wedge (FC[keyArray(k)] == FC[keyArray(j)])$

if $RFlag$ **then**

$flagArray[j] \leftarrow -1$

$FM[keyArray(j)] \leftarrow keyArray(k)$

for all $p \in keys(IL)$ **do**

if $IL[p] \in keys(FM)$ **then**

$IM[p] \leftarrow FM[IL[p]]$

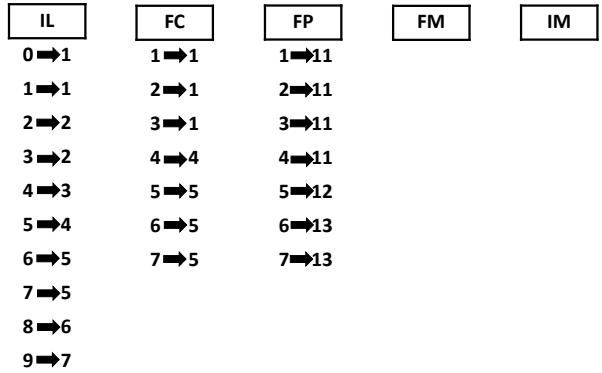
else

$IM[p] \leftarrow IL[p]$

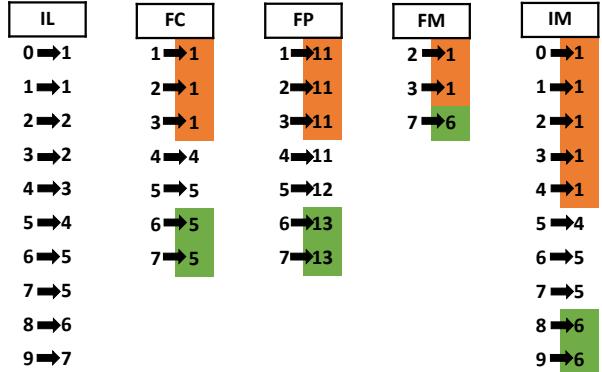
Output: IM

2 Discussion on Parameters for the Concavity Criterion

In this section we provide additional experimental illustration of the influence of parameter α_t on the number of inter- and intra-object connections. For this, we consider three high-curvature regions: concave, convex and ambiguous (see Fig. 2). We vary the value of α_t in the range 0.1 to 1.0 and analyse its influence on the classification result for the regions. The classification result is computed using eq.(2) and is provided in Table 1. One can see that for $\alpha_t = 0.1$ the convex region is classified as non-convex and removed. As such region constitutes an important region within the object, its removal negatively influences the segmentation performance and causes over-segmentation. For $\alpha_t = 1.0$ the ambiguous region (that mostly consists of measurement noise) will be classified as convex and thus preserved. Its presence will negatively influences segmentation performance and cause under-segmentation. Only for $\alpha_t = 0.2$ all three regions will be correctly classified. Therefore, for this case the segmentation performance will be the highest.



(a) Input data for the algorithm.



(b) Output data after the algorithm execution.

Figure 1: Illustration of values within input and output variables for the proposed algorithm. The merged fine labels of IM and the corresponding point groups are highlighted in color

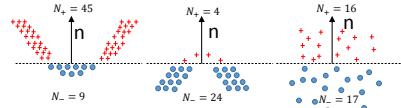
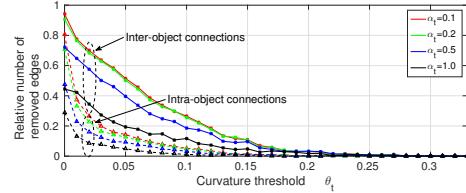


Figure 2: Illustration of high-curvature regions. Left: concave region. Middle: convex region. Right: ambiguous region

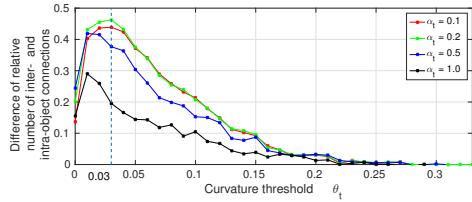
The aforementioned dependency is further illustrated experimentally. We vary the value of α_t and show the relative number of inter- and intra-object connections in Fig. 3a. One can see that for $\alpha_t = 0.2$ a significant number of inter-object connections is removed. We further see that with increasing α_t a number of removed inter-object connections is decreased. When decreasing α_t the number removed inter-object connections does not significantly change. This effect is further illustrated in Fig. 3b. Here we show the difference between relative number of inter- and intra-object connections. One can see that for $\alpha_t = 0.2$ this gain is the highest. This experimentally verifies the chosen parameter value.

Table 1: Classification result of the concavity criterion for the regions shown in Fig. 2 for various values of α_t . The correct classification result is noted as (C) and the wrong one with (F)

Value of α_t	Region type		
	Concave	Convex	Ambiguous
0.1	non-convex (C)	non-convex (F)	non-convex (C)
0.2	non-convex (C)	convex (C)	non-convex (C)
1.0	non-convex (C)	convex (C)	convex (F)



(a) Relative number of removed edges for inter- and intra-object connections



(b) Difference of relative number of removed edges for inter- and intra-object connections

Figure 3: Illustration of the influence of parameter α_t on the number of intra- and inter-object connections. This data was generated using the proposed laser scanner dataset

3 Experimental Evaluation

Here we give results for three scenes, which are omitted in the main part due to lack of space (see Fig. 4 and Fig. 5).



Figure 4: Example results for manually labelled scenes 2, 5 and 6 (left, middle and right column respectively). Here row A corresponds to images (given for illustration, but not used by any of the algorithms). B is the fine GT. C illustrates coarse GT. D represents LCCP segmentation results. E shows segmentation results of Van Kaick *et al.*. F corresponds to segmentation results of the proposed method.

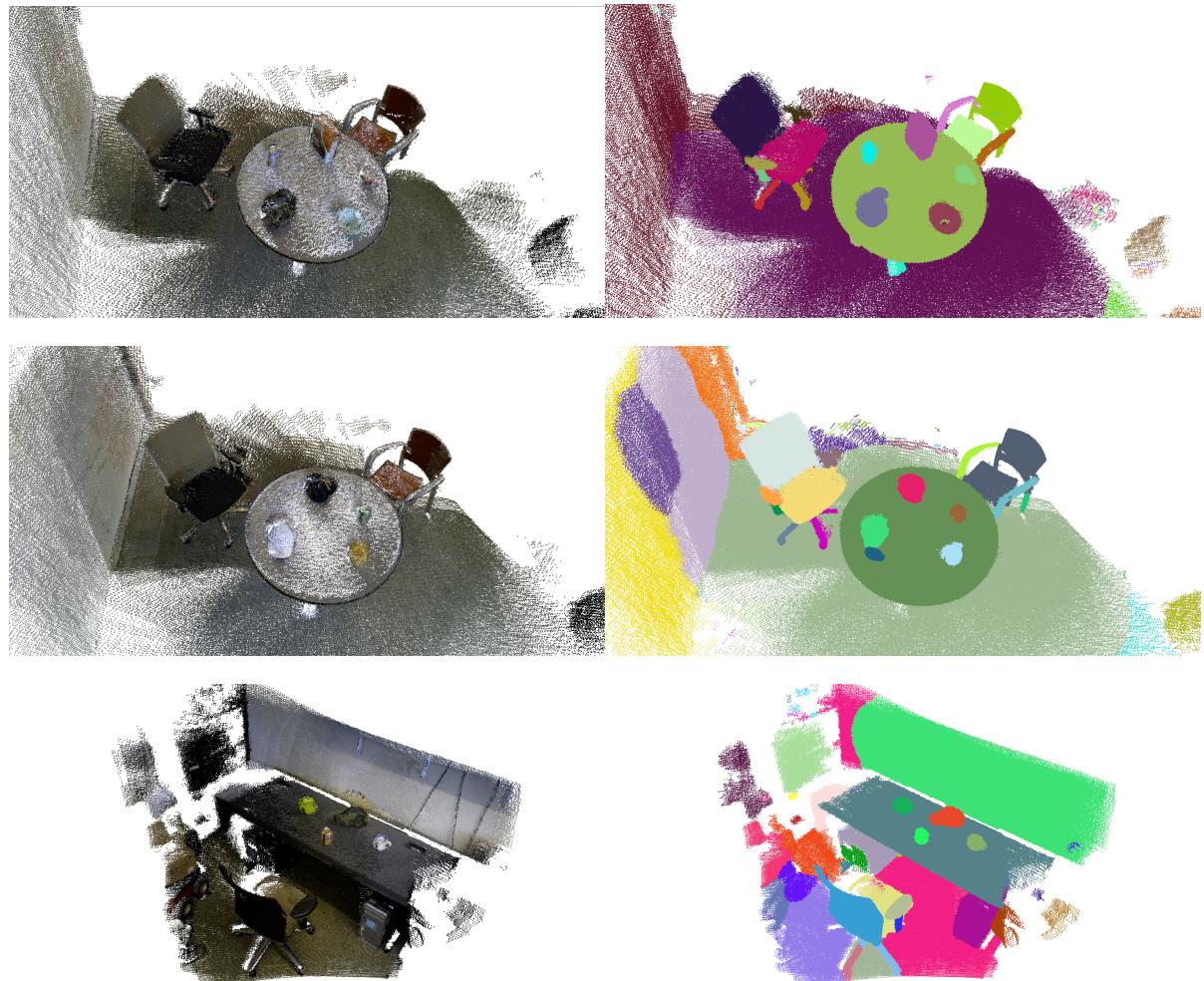


Figure 5: Example results for the dataset of Lai *et al.* From top left to bottom right: scene 6 pointcloud data and our segmentation result (zoomed in version of Fig.9 in the paper), scene 8 pointcloud data and our segmentation result and 14 pointcloud data and our segmentation result (zoomed in version of Fig.9 in the paper)