Interactive Segmentation of RGB-D Indoor Scenes using Deep Learning

Maximilian Ruethlein [1] **Franz Koeferl** [1] **Wolfgang Mehringer** [1] **Bjoern Eskofier** [1]

Abstract

Human segmentation of point clouds for the creation of datasets for deep learning is a tedious and especially time-consuming task. Interactive segmentation methods from the domain of RGB images reduce this time effort by using an iterative scheme of deep neural networks and human labelers. We apply this interactive segmentation scheme to the point cloud domain, using PointNet as our backbone and exploiting user marker-information by label propagation. The evaluation was based on a user study and suggests a significant increase in segmentation speed, which is traded off for mask quality. On average, our approach decreased segmentation time by 21.1%. We consider this work as an important first step into the domain of interactive point cloud segmentation.

1. Introduction

Point clouds are used in a variety of domains, from large scale scans in geomorphic analysis, vegetation and urban monitoring, to applications on smaller scales like self-localization and environment scanning in autonomous driving or indoor-/object scanning and reconstruction for VR applications (Otepka et al., 2013; Xie et al., 2020). Inexpensive sensor systems, like the Microsoft Kinect (Jungong et al., 2013), opened the field of point cloud scanning to the end-consumer market. With the developments in hardware and the resulting increasing availability of data, the research of processing techniques for point clouds gained traction in the deep learning community (Su et al., 2015; Maturana & Scherer, 2015; Qi et al., 2017a).

[1]Machine Learning and Data Analytics Lab, Friedrich-Alexander-University of Erlangen-Nuremberg, Erlangen, Germany. Correspondence to: Maximilian Ruethlein <maximilian.ruethlein@fau.de>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

Deep neural networks approximate arbitrary functions, finding application in a wide range of problems, but are prone to overfitting. To avoid this, the parameter count has to be matched with an appropriately sized set of labeled samples. To this end, the scientific community has created many datasets and benchmarks that can be used for training and evaluating new approaches (Deng et al., 2009; Geiger et al., 2012; Lin et al., 2014; Hackel et al., 2017).

Creating labeled datasets for training requires human annotators. Annotating big sets of data is therefore expensive and time-consuming. For this reason, ongoing effort is put into minimizing the demand of human intervention by optimizing the processes and increasing the efficiency in which user-provided data is used. One possible way is to create an interactive processing pipeline, in which a human corrects potentially erroneous predictions of a segmentation network. This was recently successfully applied to the task of image segmentation (Benenson et al., 2019).

To the best of our knowledge, similar deep-learning based approaches were not yet applied in the domain of point cloud annotation. Here, interactive annotation strategies still mainly focus on traditional annotation-/selection methods, like for example using screen-space bounding-box- or polygon-selection (Cignoni et al., 2008; Hitachi Automotive And Industry Lab, 2019; CloudCompare Community, 2019). In general, working on 3D data on a 2D screen introduces extra complexity to the labeling task, which makes efficient workflows, like proposed in the domain of RGB segmentation, additionally desirable.

We propose a data-based, interactive method for instance-based point cloud segmentation. To show the flexibility of the approach, training is performed in a class-agnostic fashion. We evaluate our pipeline in a user study by comparing to a traditional annotation tool, with a main focus on segmentation quality, -speed and minimality of user intervention. To this end, we implemented both functionalities in a custom UI. We show that our pipeline is able to use provided user information to improve segmentation masks and does significantly decrease segmentation time. However, in the current form the pipeline does not yield the same quality as a human labeler does. We therefore do not yet consider our approach as an alternative to the existing tradi-

tional workflows, but a baseline we strive to improve upon. Hence, this work gives insight in a possible realization of an interactive pipeline for point cloud segmentation.

1.1. Related Work

Interactive segmentation methods have been studied extensively in the past, with a heavy focus on image processing (Kass et al., 1988; Mortensen & Barrett, 1995; Jianbo Shi & Malik, 2000; Boykov & Jolly, 2001; Rother et al., 2004; Grady, 2006; Bai & Sapiro, 2007; Gulshan et al., 2010). Early interactive point cloud segmentation methods often directly adapted successful approaches from image processing, like graph-cuts (Li et al., 2004; Liu & Boehm, 2014).

Aforementioned approaches share their strong dependence on local, low level features for segmentation. In many cases, this is insufficient when distinguishing the object from its background, making excessive user interaction necessary. Consequently, deep-learning based approaches were soon incorporated into interactive pipelines.

For image segmentation, Xu et al. (2016) were one of the first to directly use user-input in a deep learning segmentation network by the usage of euclidean distance maps created from the user input. Further publications investigated the usage of gradient-based geodesic distance maps (Wang et al., 2019) or the usage of super-pixels (Majumder & Yao, 2019) for a better content awareness. Recently, Benenson et al. (2019) conducted a large scale study exploring interactive annotations, producing 2.5 million instance masks.

Directly learning on point cloud data is a relatively new, but very active area of research. Early segmentation approaches, using deep learning transformed the data into regular formats (Su et al., 2015; Maturana & Scherer, 2015). The pioneering work of Qi et al. (2017a) introduced PointNet as a way to learn directly from point cloud data by the use of symmetric functions. Since then, a variety of new architectures were proposed (Qi et al., 2017b; Engelmann et al., 2017; Wang et al., 2018; Landrieu & Simonovsky, 2018; Li et al., 2018; Su et al., 2018).

Deep learning approaches – to the best of our knowledge – were not yet applied to interactive point cloud segmentation. Therefore, we propose the adaption of interactive image segmentation pipelines to the domain of point clouds.

2. Segmentation Pipeline

The proposed interactive segmentation pipeline is a two-stage process. The first stage proposes an initial point-mask, segmenting the object in the center from its surrounding clutter, i.e. walls, floor, etc. Due to inexact labels in the training data and the class-agnostic prediction of the samples, the proposed mask of the first stage is expected to contain errors. In the second stage, a human is indicating errors by placing 3D markers in error regions. This information is used to update the mask, yielding a new proposal. After that, based on the new mask, further corrections can be given in multiple correction-proposal rounds.The overall process is visualized in figure 1.

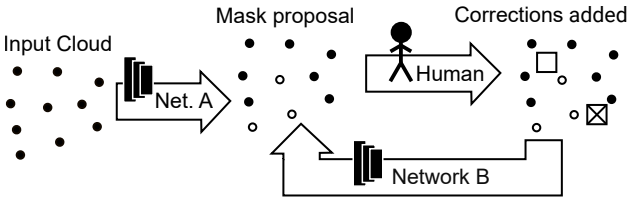

Figure 1. Overview of the interactive segmentation pipeline. Given a point cloud sample, network *A* proposes an initial segmentation mask (non-filled points). A human annotator adds 3D corrections indicating false-positive or -negative errors (empty/crossed squares). Given the corrections, network *B* improves the prediction.

In section 2.1 the extraction of training samples is presented, followed by details for the two stages of our pipeline in sections 2.2 and 2.3. Pipeline training and evaluation-metrics are described in sections 2.4 and 2.5.

2.1. Training Data

We used the SUN RGB-D dataset (Song et al., 2015) as a basis for sample extraction. It contains 10k RGB-D images of various indoor scenes captured with four different sensor systems, annotated on a per object basis. For training and testing of the separate stages of our pipeline, object-instance based point clouds were extracted from the RGB-D images.

SUN RGB-D offers labels for about 800 object categories, from which we selected ten for the feature extraction. Starting from the 37 most common classes, we removed stuff-cases (e.g. wall, floor, ceiling) and excluded categories in which the mean pixel-amount per cloud fell below 128 points. Furthermore, classes were not considered, if more than 5% of the annotation polygons covered more than 20% of the total RGB-D image. After this 16 classes remained, from which we sampled randomly.[1]

The dataset offers independent 3D bounding box and 2D polygon labels for the scene objects. Due to their greater amount and precision, we opted for using the 2D polygons. Analogous to Qi et al. (2018), point cloud samples were extracted based on camera-aligned frustums. The bounding box for the annotation polygon was extruded into the scene, capturing the object and surrounding clutter. Likewise, the

[1]Note that the remaining six classes were used for evaluating class-agnostic prediction in section 4.1.

Table 1. Total counts of extracted samples for selected object classes. Bold printed classes were selected for pipeline training. Remaining ones were used to demonstrate class agnostic prediction (section 4.1.2).

CHAIR	7366	**DRAWER**	672
CUP	392	**TRASHCAN**	264
BOX	1570	PAPER	327
PLANT	252	PICTURE	363
LAMP	887	SIGN	67
PILLOW	2072	SHELF	104
COMPUTER	266	BOOKS	324
MONITOR	527	KEYBOARD	58

segmentation mask was created by extruding the annotation polygon. Extracted samples were normalized by rotating the points so that the center of the bounding box matched the principal ray of the camera, mean-centering and scaling by standard-deviation.

In a last filtering phase, we first removed samples containing less than 256 points. Due to inaccuracies in the polygons, in some cases neighboring objects were unintentionally included in the mask, which introduced big jumps in the depth. If the depth exceeded the minimal bounding-box width by a factor greater than five, the sample was also not considered.

We derived our test-train split from the official test-train splits from the dataset. We than further split the samples into two subsets for each of the pipeline stages. Table 1 gives an overview of the total amount of extracted samples per object category.

2.2. Initial Segmentation Network

The initial segmentation model *A* was implemented using the PointNet architecture (Qi et al., 2017a). As the network uses a fixed input size, 1024 points were randomly subsampled. Afterwards, the prediction was redistributed to the original points using a distance-weighted k-nearest neighbor classifier with $k = 3$.

We converted the RGB values of the points to a single luminance value l using the BT.601 luma conversion $L = 0.299R + 0.587G + 0.144B$. This standardized conversion was used, as it is based upon the sensitivity of the human vision to each of the color channels of RGB (Poynton, 2003) and hence corresponds to the task of interactive segmentation. We observed that using luminance over RGB increases mask accuracy by 2%. The input for network *A* is therefore:

$$A_{in} = \{(X_i, Y_i, Z_i, L_i)\}, \quad \forall i \in \{1, \dots, N\}$$

2.3. Interactive Segmentation Network

The interactive segmentation network *B* improves a proposed mask M_p, that was either created by network *A* or by network *B* in a previous round. Our round based correction scheme is characterized by two quantities: The amount of clicks that can be given in a single round C and the amount of rounds itself R. In the following, we denote different round-click-configurations as $R{\times}C$. For our pipeline, we used a $3{\times}3$ setup.

We implemented network *B* as an extension of network *A*, inheriting the architecture and input size. We extended the input by M_p and a per-point input M_c, generated from the aggregated 3D markers placed by the human annotators over the course of the correction rounds. The input for network B is:

$$B_{in} = \{(A_{in;i}, M_{p;i}, M_{c;i})\}, \quad \forall i \in \{1, \dots, N\}$$

Correction Propagation The per-point correction mask M_c was generated by propagating a correction's label e_j to nearby points of the point cloud. The label indicates if the corrected error is of type false-positive (1) or false-negative (-1). For each sample point p_j, the distance to the closest correction c_j was thresholded using ϵ to determine if the label should spread:

$$M_{c;i} = \begin{cases} e_j, & \text{if } \|c_j - p_j\|_2 < \epsilon \\ 0, & \text{else} \end{cases} \tag{1}$$

In addition, we assessed a Gaussian based propagation in which weights were assigned based on the inverse distance to the correction. This was dismissed after a 5% performance loss was observed.

Click Simulation Training network *B* required corrective clicks. As training with human input is not feasible, this step was approximated by a click simulation. Given the current mask proposal M_p and the actual segmentation ground truth from the dataset, the current set of wrongly classified points was extracted. From these, *C* point positions were selected randomly. For the final correction c_j, each sampled point p_j was perturbed by a random vector and combined with the error label e_j:

$$c_j = (p_j + \mathcal{N}(0, 0.1\epsilon), e_j), \quad \forall j \in \{1, \dots, C\} \tag{2}$$

We also investigated training using an approach based on unsupervised spectral clustering of the error points. It was rejected after yielding worse results.

2.4. Pipeline Training

We used the PointNet implementation of the Kaolin PyTorch Library (Jatavallabhula et al., 2019) for our experiments and trained on a Nvidia Titan Xp. Both segmentation networks were trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 8. For network *B* a correction

spread distance of $\epsilon = 0.025$ was used. The task was set up as a per-point two-class classification problem (object=1 and clutter=0). The optimization loss was binary cross entropy.

Network *A* was trained directly on the extracted samples of the according sub-split. After training was finalized, network *A* was used for generating samples for network *B*, which used *A*'s mask proposal as an input feature.

For training network *B*, we found that a high amount of rounds also improved segmentation quality in early correction rounds. Because of this, the network was trained as a 25×3 random click setup, although it was used as a 3×3 one during the study. For each of the simulation rounds, we re-sampled 1024 new points from the point cloud sample.

2.5. Evaluation Metrics

We evaluated the per-point accuracy, recall and precision of the masks. Additionally, we used the IOU score (Jaccard, 1912) for comparison, because the object/clutter ratio was unbalanced (60:40). As shown in table 1, object classes were heavily unbalanced. To avoid the evaluation focusing on the most frequent classes, metrics were reported as the mean of the per-class means.

The simulation of clicks for network *B* was based on the selection of random points that were classified incorrectly (section 2.3). To get a more stable evaluation, we predicted each sample five times and averaged the results.

We furthermore analyzed the development of the segmentation masks over the course of the simulated correction rounds. *Self-agreement* was calculated as the IOU over the five masks that resulted from the separate runs per sample, to gain insight into the effect of varying click positions and the resulting deviations in the correction masks. Analogously, *cross-agreement* was calculated over the generated masks for one round and the masks of the previous round as a measurement of the rate-of-change in the masks.

3. Study Design

The proposed annotation pipeline was evaluated in a user study. The study followed a repeated measures design. To this end, participants labeled a set of samples, both *manually* using a polygon selection tool and *guiding* using the proposed approach. Both methods were implemented in a custom segmentation tool.[2] Segmentation quality over time was measured and logged.

Participants were searched in the age range between 20 and 40. As the study focused on novice point cloud labelers, prior knowledge for this task was an exclusion criterion.

[2]See supplementary material for more information on the tooling and detailed workflows.

The study was conducted on 13 test-samples of network *B*. Samples were selected in a way that the range of difficulty was reflected. In the KITTI benchmark (Geiger et al., 2012) samples' bounding box sizes were used as an indicator for difficulty.[3] Similarly, we sampled over the range of point cloud sizes. To mitigate learning effects from the sample being shown once for each method, the sample-approach-*combinations*' order was randomized for each subject. This also reduced effects of fatigue.

3.1. Activities

In the introduction phase, questionnaires were conducted[4], the tools' controls were explained and subjects trained on a set of practice-samples. After the introduction, participants worked independently on the combinations.

For the guiding approach, up to three clicks in three rounds were provided. To make full use of the extrapolation capabilities of network *B*, participants were instructed to mark the center of error volumes, starting with the biggest ones. They were free to give fewer corrections if they felt the current markers were already sufficient, or no further corrections if they were content with the mask. In the latter case, remaining rounds were skipped and the next combination was displayed.

For the manual approach no direct external limit (i.e. restricted clicks and rounds) for possible user actions was given. Attendants were instructed to refine the mask until they were satisfied with the result. For extreme cases, an extensive time limit of 10 minutes was put in place.

After all combinations were finished, another questionnaire was conducted focusing on usability aspects, assessing possible problems and shortcomings of the tool that might have interfered with the process. An overview of the major study steps is given in figure 2.

3.2. Comparison of Methods

The proposed approach was evaluated in terms of the mean relative per-participant speed difference between the two methods, in relation to the segmentation quality. Due to artifacts and inaccuracies, a direct comparison based on the SUN RGB-D dataset's ground-truth was not possible, as many of the point labels did not match the human intuition of object boundaries in 3D space. The network on the other hand learned on data containing this error and might have been able to reproduce them, leading to distorted re-

[3]In addition to bounding box height, in the KITTI dataset occlusion and truncation were used to distinguish hard and easy cases.

[4]See supplementary material.

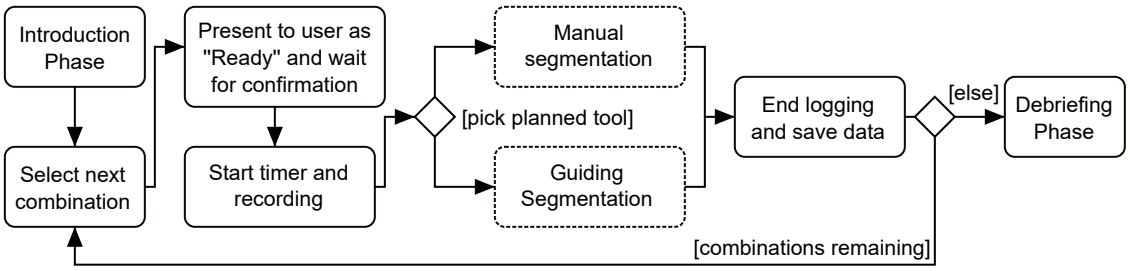

Figure 2. Overview of the major steps of the conducted study. After the introduction phase, a sample-tool combination was selected based on the randomized order for the current participant. When the participant was ready, the segmentation with the current tool started. In parallel, time and user interactions were logged. After finishing the sample, the user either continued with the next combination, or entered the debriefing stage.

Table 2. Training results of the pipeline's segmentation networks. Configurations marked with *uc* were evaluated on unknown classes. Reported values for network *B* were measured after the third simulation round.

	IOU	ACC.	REC.	PREC.
NET. A	67.0	74.3	85.2	80.3
NET. B	75.5	83.6	86.7	84.4
NET. A (UC)	64.6	69.4	86.6	67.3
NET. B (UC)	74.9	81.8	86.3	84.8

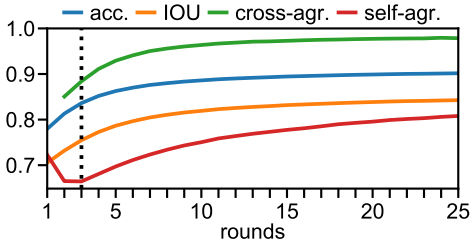

Figure 3. Development of evaluation metrics for network *B* over progressing, simulated correction rounds. Values for round 3 are marked by the dotted line.

sults. Instead, the ground truth for one specific sample was derived by averaging all masks created with the manual segmentation tool throughout the study.

In addition, the segmentation process was evaluated by tracking quality over time. To be able to compare and average all times from all participants, the manual segmentation time was used for normalizing. This way, different absolute work speeds did not influence the time comparison between both tools.

4. Results & Discussion

We present our results in two parts. In section 4.1 we evaluate the pipeline based on simulated clicks. Section 4.2 describes results of the user study.

4.1. Simulation Results

We evaluated our pipeline using samples from classes selected for training (section 4.1.1) and additionally a set of classes that were unknown to the pipeline (section 4.1.2). An overview of the evaluation scores is given in table 2.

4.1.1. Network Performances

Both networks were trained separately. We trained network *A* over 343 epochs before overfitting set in, reaching an IOU of 67.0%.

Network *B* was trained over 147 epochs. As mentioned in section 2.4, we used a 25×3 setup with simulated, random clicks. Note that due to every batch being simulated over 25 rounds, each epoch contained approximately 25 times more optimization steps than for network *A*. At round three, an IOU of 75.5% was measured. Given the full set of 75 cumulative clicks after 25 rounds, 84.2% IOU was reached.

Figure 3 shows the development of evaluation metrics over the simulation rounds. We observe a steady increase in mask quality. The measured self-agreement of the network predictions indicate that predicted masks converged towards a similar solution, even if the correction-positions and -orders were varied. The growing cross-agreement showed that subsequent mask proposals became increasingly similar, i.e. every additional round produced a smaller deviation from the current mask.

When run over an extensive amount of simulated correction rounds, IOU scores >80% were reached. Therefore, corrective information was successfully used by the pipeline. However, the quality gained per round was low and decreased with progressing rounds.

4.1.2. Predicting Unknown Classes

Additionally, we evaluated the generalizability of our approach to classes not seen in the training phase. For this, the remaining six classes of the class-selection process (sec-

tion 2.1) were used. For network *A* the mean segmentation IOU was 64.6%. After running network *B* for three correction rounds, the overall IOU of the predictions rose by 10.3% to a total of 74.9%.

This indicates network *A* operating on the classes that have been used for learning is beneficial, which is expected. On the other hand the gap in IOU between evaluating on known and unknown classes is relatively small (2.4%) and became even smaller (0.6%) after applying the correction network *B*. Therefore, we conclude that the overall pipeline does offer class-agnostic segmentation. Network *B* forms the major factor, because the regularization effect of the (simulated) user clicks and the sense of locality given by the resulting correction mask might have promoted the learning of more general features.

Note that good initial mask proposals still are important, as the mask is passed to the interactive segmentation network as an input. Reported high cross-agreement values over successive correction rounds (section 4.1.1) indicate that mask updates happen in relatively small iterations. Therefore, a major error introduced by a faulty initial configuration might slow or prevent convergence towards the solution.

4.2. Study Results

The study group consisted of 9 males and 7 females aged from 21 to 33 years with a mean age of 26.2 years. The segmentation-runs lasted between 43 and 93 minutes, with a mean duration of 65 and a median of 64 minutes.

4.2.1. Qualitative Analysis

Figure 4 gives examples for the masks created during the study. We show how many of the participants labeled a point as part of the object by using color coding. Varying colors in an area of the sample indicate disagreement among the masks created by the subjects.

For the manual masks, all subjects labeled the major structures of the scene in a similar fashion, with low noise towards the edges. Fine structures, like chair legs (example c) or the leaves of a plant (example d) showed an increased amount of disagreement. Masks resulting from the guiding approach showed a noticeably higher amount of noise. The areas of high agreement roughly followed those of the manual masks, but showed strong noise towards the object borders. Like for the manual masks, fine structures posed a challenge, with the mask for the plant sample (example d) only very roughly following the shape of the object.

4.2.2. Manual Mask Agreement

Study evaluation was based on the manual segmentation masks of the participants, combined by majority-voting for each point of a sample (section 3.2). To verify that this is sensible, we evaluate the general quality of the manual masks. Following the approach of Hackel et al. (2017), pairwise IOU scores of all participants and samples were calculated for the manually created masks. A high agreement throughout the subjects was considered an indicator for good mask-quality.

Figure 5 shows the distribution of the mean IOU over all samples for each of the subject-pairs. Per-pair IOU scores primarily clustered around 80%, with a second distinct cluster center at roughly 68%, resulting from one subject differing noticeably from the remaining study participants. The cluster for the diverging subject and the remaining pairs were each found to be normal-distributed using the Shapiro-Wilk test (Shapiro & Wilk, 1965) (α=0.05, p=0.99 and p=0.81, respectively). Comparing the clusters using a t-test reported a significant difference for the IOU values (α=0.05, p<0.001), i.e. the subject differed from the general intuition about the locality of the objects. This indicates, that the participant either did not fully understand the task at hand or did not work conscientiously. For this reason, the subject was declared an outlier and excluded from further evaluations. After exclusion, the remaining pairs showed a mean IOU of 80% with minimum and maximum values of 73.2% and 88.7%, respectively.

To set this into context, the PASCAL VOC object detection benchmark (Everingham et al., 2010) uses the IOU score threshold of 50% as a measure to determine correct detections. They reported this threshold was "set deliberately low to account for inaccuracies" and is hence seen as a lower bound for acceptable IOU scores in this work. For the Semantic3D.net benchmark (Hackel et al., 2017) pairwise annotator agreements were assessed, by calculating IOU scores from overlapping areas. They reported agreement values >95% IOU. User-agreements for our study surpassed the 50% threshold, but did not reach values over 90%. Hackel et al. used LIDAR-based point clouds in an urban setting, labeling major structures like man-made terrain, natural terrain, etc. We argue that labeling indoor RGB-D data, whose acquisition modality offers less accuracy (Xie et al., 2020), is more challenging.

In addition, we directly compared the majority-voted user masks to the labels extracted directly from the SUN RGB-D dataset. In spite of deviations due to label noise, we expected an overall high agreement. IOU sample scores reached from 54.2% to 95.5% with a mean of 81.6% and a median of 87.4%. When comparing based on accuracy, values ranged from 73.6% to 96.5%, with a mean of 88.5% and a median of 91.0%.

Nevertheless, some user masks did diverge from the original ground truth with an IOU <60%: In examples d) and f) in figure 4 users were asked to segment plants in low-resolution areas of the image, making distinction difficult. In addition,

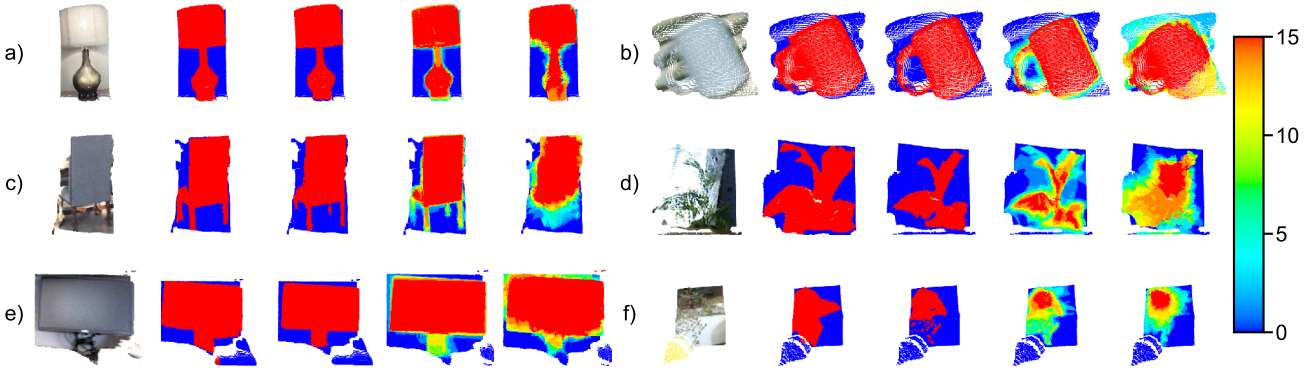

Figure 4. Examples for study samples and results. Each set shows from left to right: RGB-D point cloud, SUN RGB-D ground-truth, user created ground-truth (majority vote from manual masks), summed labels from the manual approach, summed labels from the guiding approach. For the summed labels, the color represents how often a point was labeled as an object during the study.

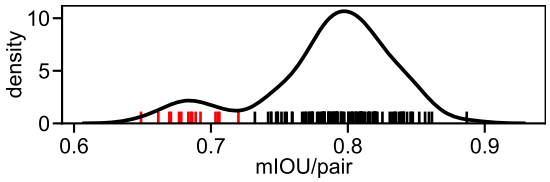

Figure 5. Density of pairwise manual user mask agreement, averaged over all samples. Ticks on the x-axis give the discrete measurements, with the red ticks marking pairs originating from the excluded subject.

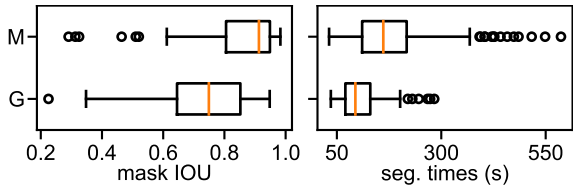

Figure 6. Distribution of IOU and segmentation time in seconds over all participation runs of the study, grouped by the approach (M=manual, G=guiding).

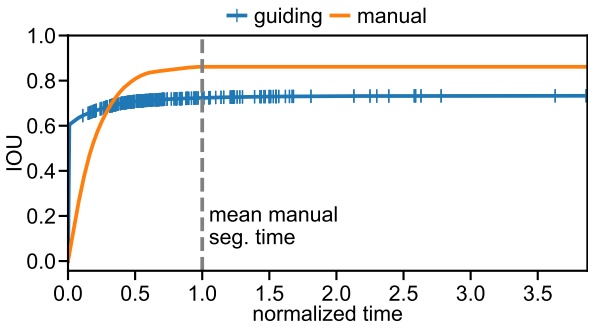

Figure 7. Average IOU per tool over normalized time. Vertical line-ticks mark end-times of guiding annotation runs.

the dataset-based plant-masks were extracted from rough polygons, which did not accurately follow the objects' borders. This mismatch between the labeling modalities of the SUN RGB-D dataset (polygons) and the implemented manual labeling tool (per-point labels) is an additional cause for lower IOU scores. For example in b), the inside of the cup handle was included in the polygon, while it was successfully cut out during the study. Nonetheless, visual inspection of created masks shows that most user masks do follow or even improve the dataset ground-truth. Following the approach for the pairwise agreements, given the mean IOU of 81.6%, we conclude that the mean user masks are reasonably similar to the original dataset.

Given the aforementioned factors, we state that subjects did follow a common, correct intuition, making comparing against the overall mean mask sensible.

4.2.3. Speed and Quality Comparison

Figure 6 shows the distribution of all participants' segmentation IOU scores and times for all samples per approach. Using the Shapiro-Wilk test (α=0.05), segmentation times were found to be log-normal (Choi, 2016), while segmentation quality measures did not follow a normal distribution. Given those results, log-segmentation-times were compared using a two-sided t-test (Student, 1908). For the mask IOU scores the Wilcoxon signed-rank test (Wilcoxon, 1945) was used. We found significant differences for both the log-times (p<0.001) and IOU scores (p<0.001), i.e. our approach significantly improved segmentation time, but reached lower IOU scores.

IOU over Time Mask quality was logged throughout the subjects' participations. We normalized every annotation-run's duration by using the respective manual segmentation time (section 3.2). This way the average mask quality development over time for each of the tools was analyzed (figure 7).

Both approaches' IOU was 0% at the beginning, as the study's start configuration was set to an empty mask, i.e. all points were labeled as "clutter". Guiding IOU quickly rose to ~60% from the initial prediction of network *A*. From there, network B only slowly increased mask quality to a

final 73.3%. Compared to this, manual IOU rose distinctly faster, while also reaching a higher final quality of 86.1%. Both curves intersected after 31% of average manual segmentation time, at an IOU of 69%.

End-times of the guiding runs spread from 11.5% to 386% of their respective manual time. Overall, mean relative duration for guiding runs was 78.9%, i.e. segmentation with the guiding approach on average took 21.1% less time than with the manual one. In total, 75.4% of all samples were finished faster using the proposed approach.

The development of IOU over time showed that, equally to the training samples created using simulated clicks (section 4.1), corrective information given by the subjects did improve mask quality. Compared to the manual results, mask quality did not reach sufficient values. We showed that the proposed pipeline led to significantly shorter annotation times, with a majority of the samples being completed faster than using the traditional approach. However we want to note, that while the manual approach was not bounded, the guiding approach did impose an upper limit in terms of clicks and rounds. This was done to enforce the minimality of user intervention and to follow previous work on the topic (Benenson et al., 2019), but also introduced a prior towards lower segmentation times.

From this we conclude that even though segmentation times were improved, the overall extrapolation capabilities of the presented interactive segmentation network *B* are not sufficient. Due to this, the pipeline is not considered an alternative to traditional segmentation workflows in terms of mask quality.

4.2.4. User Interactions

As initially stated, the proposed segmentation pipeline was aimed at being minimal in terms of user intervention. This was examined by counting central interactions that were done by the subjects during the study. For the manual segmentation approach, the creation of a new point-selection using the polygon selection tool was considered the central operation. For the guiding segmentation approach the additions of new corrections to the scenes were counted.

For all manual segmentation a mean of 7.14 and a median of 5 interactions per sample were measured with the minimum and maximum being 1 and 35, respectively. For the guiding approach a mean of 9.47 and a median of 9 interactions were counted. Guiding interactions ranged from 2 to 18. Given the 3x3 configuration of the proposed pipeline, an average interaction count close to 9 was expected.

For both approaches, the expected workflows contained more interactions than the ones previously stated, like for example setting the mask-/correction-label or moving the camera. They were not counted as central operations, as they were realized analogously for both workflows and therefore did not influence our comparison.

Simply counting interactions omits their respective complexities. Placing a correction marker in the scene required the subjects to click error clusters once. In contrast, the creation of a manual selection by drawing a polygon requires at least three clicks to span an area. Multiplying the aforementioned mean interaction-counts for the approaches with a respective weighting therefore gives $7.14 \times 3 = 21.42$ for the manual and 9.47 clicks for the guiding approach, assuming only minimal polygon-selections of vertex count 3. Hence, taking individual mouse clicks into account, the guiding segmentation approach needed at most half as many clicks as the manual one.

Obtained results are highly dependent on the way the segmentation tools were implemented during the study. Yet, it is plausible to assume that results will not differ greatly when implemented in a more advanced UI, because guiding segmentation requires single points, whereas manual segmentation on a 2D screen still requires the creation of areas. We conclude that the approach used in the study was minimal in terms of user intervention.

5. Conclusion & Future Work

We evaluated an interactive, data-based segmentation approach for RGB-D indoor point clouds in which a user corrected the predictions of a segmentation networks over multiple correction rounds. We showed that the interactive stage of our pipeline was able to bridge the gap to unknown classes given minimal user input, i.e. the approach can be run in a class-agnostic fashion. For quality and speed analysis, a user study was conducted to directly compare our method with a traditional one. Our pipeline was able to gain segmentation mask quality given user input and to decrease segmentation time. However the quality gain per correction round was small and the method did not reach values better or equal to the traditional approach.

We used the PointNet architecture as a well known and tested baseline. Recent research for deep learning on point clouds proposed a multitude of new architectures more sensitive to the local spatial contexts inside the point cloud (Qi et al., 2017b; Li et al., 2018; Wang et al., 2018). We expect that the application of an according approach will increase the positive effect of the user input on the final masks and therefore promote the adaptation of the pipeline.

The success of interactive approaches in the domain of image processing and the ongoing research considering deep learning on point cloud data makes us confident that similar approaches will achieve satisfactory results in the near future. Our work is a first step in this direction and forms a baseline for future work.

References

Bai, X. and Sapiro, G. A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting. In *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8. IEEE, 2007.

Benenson, R., Popov, S., and Ferrari, V. Large-scale interactive object segmentation with human annotators. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11700–11709, 2019.

Boykov, Y. and Jolly, M.-P. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pp. 105–112. IEEE, 2001.

Choi, S. W. Life is lognormal! What to do when your data does not follow a normal distribution. *Anaesthesia*, 71 (11):1363–1366, 2016.

Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., and Ranzuglia, G. MeshLab: An open-source mesh processing tool. In *6th Eurographics Italian Chapter Conference 2008 - Proceedings*, pp. 129–136. The Eurographics Association, 2008.

CloudCompare Community. CloudCompare (version 2.6), 2019. URL http://www.cloudcompare.org/.

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

Engelmann, F., Kontogianni, T., Hermans, A., and Leibe, B. Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, volume 2018, pp. 716–724. IEEE, 2017.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

Geiger, A., Lenz, P., and Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361. IEEE, 2012.

Grady, L. Random Walks for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.

Gulshan, V., Rother, C., Criminisi, A., Blake, A., and Zisserman, A. Geodesic star convexity for interactive image segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3129–3136. IEEE, 2010.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., and Pollefeys, M. Semantic3D.net: A new Large-scale Point Cloud Classification Benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 91–98, 2017.

Hitachi Automotive And Industry Lab. Semantic Segmentation Editor, 2019. URL https://github.com/Hitachi-Automotive-And-Industry-Lab/semantic-segmentation-editor.

Jaccard, P. The Distribution of the Flora in the Alpine Zone. *New Phytologist*, 11(2):37–50, 1912.

Janoch, A., Karayev, S., Jia, Y., Barron, J. T., Fritz, M., Saenko, K., and Darrell, T. A Category-Level 3D Object Dataset: Putting the Kinect to Work. In *Consumer Depth Cameras for Computer Vision*, pp. 141–165. Springer London, London, 2013.

Jatavallabhula, K. M., Smith, E., Lafleche, J.-F., Tsang, C. F., Rozantsev, A., Chen, W., Xiang, T., Lebaredian, R., and Fidler, S. Kaolin: A PyTorch Library for Accelerating 3D Deep Learning Research. *arXiv*, 1911.05063, 2019.

Jianbo Shi and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Jungong, H., Ling, S., Dong, X., and Shotton, J. Enhanced Computer Vision With Microsoft Kinect Sensor: A Review. *IEEE Transactions on Cybernetics*, 43(5):1318–1334, 2013.

Kass, M., Witkin, A., and Terzopoulos, D. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

Landrieu, L. and Simonovsky, M. Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4558–4567. IEEE, 2018.

Li, Y., Sun, J., Tang, C.-K., and Shum, H.-Y. Lazy snapping. *ACM Transactions on Graphics*, 23(3):303, 2004.

Li, Y., Rui, B., Sun, M., Wu, W., Di, X., and Chen, B. PointCNN: Convolution On X-Transformed Points. In *Advances in Neural Information Processing Systems*, pp. 820–830, 2018.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common Objects in Context. In *Lecture Notes in*

Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 8693 LNCS, pp. 740–755. 2014.

Liu, K. and Boehm, J. A New Framework For Interactive Segmentation of Point Clouds. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-5(5):357–362, 2014.

Majumder, S. and Yao, A. Content-Aware Multi-Level Guidance for Interactive Instance Segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2019-June, pp. 11594–11603. IEEE, 2019.

Maturana, D. and Scherer, S. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928. IEEE, 2015.

Mortensen, E. N. and Barrett, W. A. Intelligent scissors for image composition. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques - SIGGRAPH '95*, pp. 191–198, New York, New York, USA, 1995. ACM Press.

Otepka, J., Ghuffar, S., Waldhauser, C., Hochreiter, R., and Pfeifer, N. Georeferenced Point Clouds: A Survey of Features and Point Cloud Management. *ISPRS International Journal of Geo-Information*, 2(4):1038–1065, 2013.

Poynton, C. *Digital Video and HDTV Algorithms and Interfaces*. Morgan Kaufmann Publishers Inc., 2003. ISBN 978-1-55860-792-7.

Qi, C. R., Su, H., Kaichun, M., and Guibas, L. J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2017, pp. 77–85. IEEE, 2017a.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Conference on Neural Information Processing Systems (NIPS) 2017*, pp. 5105–5114, 2017b.

Qi, C. R., Liu, W., Wu, C., Su, H., and Guibas, L. J. Frustum PointNets for 3D Object Detection from RGB-D Data. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 918–927. IEEE, 2018.

Rother, C., Kolmogorov, V., and Blake, A. GrabCut. In *ACM SIGGRAPH 2004 Papers on - SIGGRAPH '04*, pp. 309, New York, New York, USA, 2004. ACM Press.

Shapiro, S. S. and Wilk, M. B. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4): 591–611, 1965.

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. Indoor Segmentation and Support Inference from RGBD Images. In *Proceedings of the 12th European conference on Computer Vision - Volume Part V*, pp. 746–760. Springer-Verlag, 2012.

Song, S., Lichtenberg, S. P., and Xiao, J. SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In *28th IEEE Conference on Computer Vision and Pattern Recognition*, pp. 567–576, 2015.

Student. The Probable Error of a Mean. *Biometrika*, 6(1):1, 1908.

Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. Multi-view Convolutional Neural Networks for 3D Shape Recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, volume 2015 Inter, pp. 945–953. IEEE, 2015.

Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M.-H., and Kautz, J. SPLATNet: Sparse Lattice Networks for Point Cloud Processing. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2530–2539. IEEE, 2018.

Wang, G., Zuluaga, M. A., Li, W., Pratt, R., Patel, P. A., Aertsen, M., Doel, T., David, A. L., Deprest, J., Ourselin, S., and Vercauteren, T. DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1559–1572, 2019.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics*, 38(5): Article 146, 2018.

Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80, 1945.

Xiao, J., Owens, A., and Torralba, A. SUN3D: A Database of Big Spaces Reconstructed using SfM and Object Labels. In *Proceedings of 14th IEEE International Conference on Computer Vision (ICCV2013)*, pp. 1625–1632, 2013.

Xie, Y., TIAN, J., and Zhu, X. Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation. *IEEE Geoscience and Remote Sensing Magazine*, 2020.

Xu, N., Price, B., Cohen, S., Yang, J., and Huang, T. Deep Interactive Object Selection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2016, pp. 373–381. IEEE, 2016.