WHICH 3D DATA REPRESENTATION DOES THE CROWD LIKE BEST? CROWD-BASED ACTIVE LEARNING FOR COUPLED SEMANTIC SEGMENTATION OF POINT CLOUDS AND TEXTURED MESHES

M. Kölle*, D. Laupheimer*, V. Walter, N. Haala, U. Soergel

Institute for Photogrammetry, University of Stuttgart, Germany -
(michael.koelle, dominik.laupheimer, volker.walter, norbert.haala, uwe.soergel)@ifp.uni-stuttgart.de

Commission II, WG II/6

KEY WORDS: Crowdsourcing, Active Learning, Multi-Modality, 3D Point Clouds, 3D Textured Meshes, Semantic Segmentation

ABSTRACT:

Semantic interpretation of multi-modal datasets is of great importance in many domains of geospatial data analysis. However, when training models for automated semantic segmentation, labeled training data is required and in case of multi-modality for each representation form of the scene. To completely avoid the time-consuming and cost-intensive involvement of an expert in the annotation procedure, we propose an Active Learning (AL) pipeline where a Random Forest classifier selects a subset of points sufficient for training and where necessary labels are received from the crowd. In this AL loop, we aim on coupled semantic segmentation of an Airborne Laser Scanning (ALS) point cloud and the corresponding 3D textured mesh generated from LiDAR data and imagery in a hybrid manner. Within this work we pursue two main objectives: i) We evaluate the performance of the AL pipeline applied to an ultra-high resolution ALS point cloud and a derived textured mesh (both benchmark datasets are available at `https://ifpwww.ifp.uni-stuttgart.de/benchmark/hessigheim/default.aspx`). ii) We investigate the capabilities of the crowd regarding interpretation of 3D geodata and observed that the crowd performs about 3 percentage points better when labeling meshes compared to point clouds. We additionally demonstrate that labels received solely by the crowd can power a machine learning system only differing in Overall Accuracy by less than 2 percentage points for the point cloud and less than 3 percentage points for the mesh, compared to using the completely labeled training pool. For deriving this sparse training set, we ask the crowd to label 0.25 % of available training points, resulting in costs of 190 $.

1. INTRODUCTION

In recent years, significant effort was put into developing and advancing automatic Machine Learning (ML) methods such as Convolutional Neural Networks (CNNs) for various data representations, as for 2D imagery (Ronneberger et al., 2015; Badrinarayanan et al., 2017) or 3D point clouds (Qi et al., 2017; Graham et al., 2018). However, lack of labeled training data greatly hinders the application of such systems. Therefore, tremendous exertion was made for establishing massive annotated data corpera such as *ImageNet* (Deng et al., 2009). Since manual annotation of about 14 million images by experts is infeasible, this dataset was mainly built up by the available workforce of individual crowdworkers on the internet. Namely, paid crowdsourcing was applied, where in contrast to volunteered crowdsourcing such as *OpenStreetMap* (Budhathoki and Haythornthwaite, 2012), individual workers are recruited through respective platforms like *Amazon Mechanical Turk* (Buhrmester et al., 2011) or *Microworkers* (Hirth et al., 2011). However, compared to annotating images of everyday life scenes, interpretation of geospatial data by non-experts (i.e., the crowd) is far more demanding due to an unfamiliar perspective (i.e., nadir-like bird view). This complexity is further intensified when focusing on 3D data, which non-experts might have never dealt with before. However, when a semantic segmentation of 3D data is desired, working directly with the original data is most reasonable in order to avoid loss of information, for instance, by projection to a lower dimensional space. Herfort et al. (2018), Walter and Soergel (2018) and Kölle et al. (2020) already demonstrated that crowdworkers are generally capable of interpreting 3D scenes discretized by Airborne Laserscanning (ALS) point clouds. However, interpretability of 3D scenes might even be improved by combining ALS data and high-resolution imagery in form of textured 3D meshes. Compared to discretized point clouds, textured meshes provide an easy-to-interpret closed 3D surface description. Such kind of multi-modality is especially supported by the simultaneous use of an ALS sensor and camera(s) in many applications (Haala et al., 2020; Cramer et al., 2018). For combining both raw measurements in accurate georeferencing, Glira et al. (2019) have proposed to run a hybrid adjustment of both sensors' output in order to exploit the individual strengths of each principle yielding in an optimal alignment of LiDAR data and imagery.

Although pursuing aforementioned approaches might simplify the interpretation of 3D data for crowdworkers, the total amount of instances which need to be annotated is still the same, since ML models typically require fully annotated training sets. One means for significantly reducing the number of necessary labels and thus costs is Active Learning (AL) (Settles, 2009). AL is an iterative process which aims to detect those instances which are sufficient for training a ML model. This means manual labeling is focused on most informative instances only. This approach of learning from a small pool of labeled points has already proven to be successful for semantic segmentation of point clouds, as demonstrated in Luo et al. (2018), Li and Pfeifer (2019), Lin et al. (2020) and Kölle et al. (2021). While to the best of our knowledge for semantic interpretation of 3D meshes no AL approaches have been proposed in literature so far, some strategies for Pas-

*Corresponding authors

sive Learning (PL) are present. Rouhani et al. (2017), Tutzauer et al. (2019) and Laupheimer et al. (2020) have shown that this task can be realized as classification of individual faces, which are simplified to their Center of Gravity (CoG). This enables the application of classification approaches utilized for point clouds such as *PointNet++* (Qi et al., 2017), SCN (Graham et al., 2018) and Random Forest (RF) (Breiman, 2001).

Within this work, we aim on establishing a hybrid intelligence system (Vaughan, 2018) where we completely remove experts from the AL loop. Our contributions can be summarized as follows: i) We enhance the AL procedure so that our RF models for semantic segmentation of the point cloud and the 3D mesh are iteratively refined in a coupled fashion. ii) Since these models learn from the crowd, we compare whether 3D point clouds or 3D textured meshes are more suitable for presenting to crowdworkers, which are addressed by usage of the *Microworkers* platform. iii) We release a new high-resolution 3D benchmark dataset incorporating both a 3D LiDAR point cloud and mesh data, which we refer to as Hessigheim 3D (H3D) henceforth.

2. METHODOLOGY

Within this section, we discuss the two main elements of our human-in-the-loop system: i) the ML part for coupled semantic segmentation of point clouds and meshes (section 2.1 & 2.2) and ii) the way we leverage human workforce of crowdworkers (section 2.3).

2.1 AL Loop for Coupled Semantic Segmentation of Point Clouds and Meshes

AL can be considered as a construct composed of three interdependent components: i) the ML model, ii) the strategy for selection of most informative points and iii) the employed oracle $\mathcal{O}$. The interaction of these components is depicted in Figure 1. We assume to have acquired an ALS point cloud and run basic post-measurement routines (e.g., alignment of strips). This dataset can be thought of as unlabeled training pool U. For initializing the AL loop, a first small training pool T is required including samples for the desired classes. T is built by the oracle $\mathcal{O}_C$, embodied by the crowd C in our case, and then induced into the point cloud branch (see Figure 1). Based on T, we can train an arbitrary classifier for semantic segmentation of the point cloud. For this, we rely on a RF classifier, which can be easily adapted for the AL setting for its pointwise functionality and since (in contrast to CNN approaches) required features can be computed one time in advance of the AL loop (features employed are discussed in section 2.2). After inference of the trained model on the remaining training set R ($R = U \setminus T$), the main objective in AL in each iteration step is to select those instances of R which have the greatest positive influence on the performance of the model and therefore justify manual labeling effort (Settles, 2009). Here, we aim on finding an intrinsic measure for determining the uncertainty of the model's predictions. One established method is to select those points where the entropy of the respective a posteriori probabilities $p(c|x)$ that point x belongs to class c is maximum (i.e., the classifier is most uncertain about these points):

$$x_E = \underset{x}{\operatorname{argmax}} - \sum_c p(c|x) \cdot \log p(c|x) \tag{1}$$

In order to alleviate severe class imbalance ALS point clouds generally tend to suffer from, we further weight the entropy scores by dynamically derived factors determined as ratio of the total number of points n_T currently present in T and the number of representatives of each class n_c at iteration step i ($w_c(i) = n_T(i)/n_c(i)$). For efficiency reasons, we aim on selecting n points in each iteration step, which often leads to the acquisition of similar points in terms of their position in feature space. To increase the diversity of selected points and hence boost the convergence of the AL loop, we proceed as proposed by Zhdanov (2019). For this we detect clusters in feature space by running a k-means clustering (Lloyd, 1982) and form a diverse sampling by choosing one instance of each cluster derived. Therefore the number of clusters k equals the number of points n selected in each iteration step. However, we need to keep in mind that non-experts (i.e., the crowd) are asked to generate labels for the selected points. We aim to evaluate whether we can ease interpretability of sampled points by further applying the method proposed in Kölle et al. (2021), denoted as Reducing Interpretation Uncertainty (*RIU*). Sampled points are often situated exactly on class borders, where the true label is ambiguous and labeling is strongly dependent on the individual class understanding. Therefore, we try to increase the distance to the class border by considering the selected point as seed point and decide to use a point within distance d_{RIU} instead when the sampling score (i.e., the weighted entropy value) of this instance is lower, assuming that a lower score is closely related to the distance to class boundary (detailed explanation can be found in Kölle et al. (2021)). Afterwards the selected points (either applying *RIU* or not) are presented to crowdworkers for labeling.

This AL loop (denoted by *red* arrows in Figure 1) is repeated for n_i iteration steps until convergence or until exhaustion of budget. So far, this approach only allows semantic segmentation of point clouds (see point cloud branch in Figure 1). If the airborne platform is equipped with imaging sensors as well, we can further process textured 3D meshes. In order to derive a coupled semantic segmentation of 3D meshes, we associate labeled points and faces of the mesh by the method proposed in Laupheimer et al. (2020), referred to as Point Cloud Mesh Association (PCMA). This procedure works as follows: i) For each face of the mesh, LiDAR points within an unbounded prism neighborhood (with shape and orientation determined by the respective face) are selected and further filtered based on their orthogonal distance to the face's triangular plane. Due to discrepancies in representations, few points and faces remain without any associated entities. ii) The class of the face is then obtained by majority vote from all class labels of associated LiDAR points.

This method allows to transfer labels from individual points of the LiDAR point cloud (set by crowdworkers) to the 3D mesh. Therefore, samples for the mesh are as well implicitly generated. Thus, we can couple both segmentation processes (see Figure 1). This consequently makes a second independent AL run for the mesh expendable and labeling effort for the meshed representation can be completely avoided by exploiting the overlaying nature of the mesh and the point cloud. However, the performance of the RF employed within the AL loop for semantic segmentation of the mesh might be diminished, since although we employ the RF classifier for both segmentation tasks, the models would actually differ due to different primitives (3D LiDAR points vs. mesh faces) and features (see section 2.2). Therefore, different positions in the two representation forms would be selected by the classifier, so that a fair comparison of which data representation is best suited for presenting to crowdworkers would not be possible. This can only be achieved within the coupled approach by visualizing context of the exact same selected point, on one hand by the point cloud and on the other hand by the textured

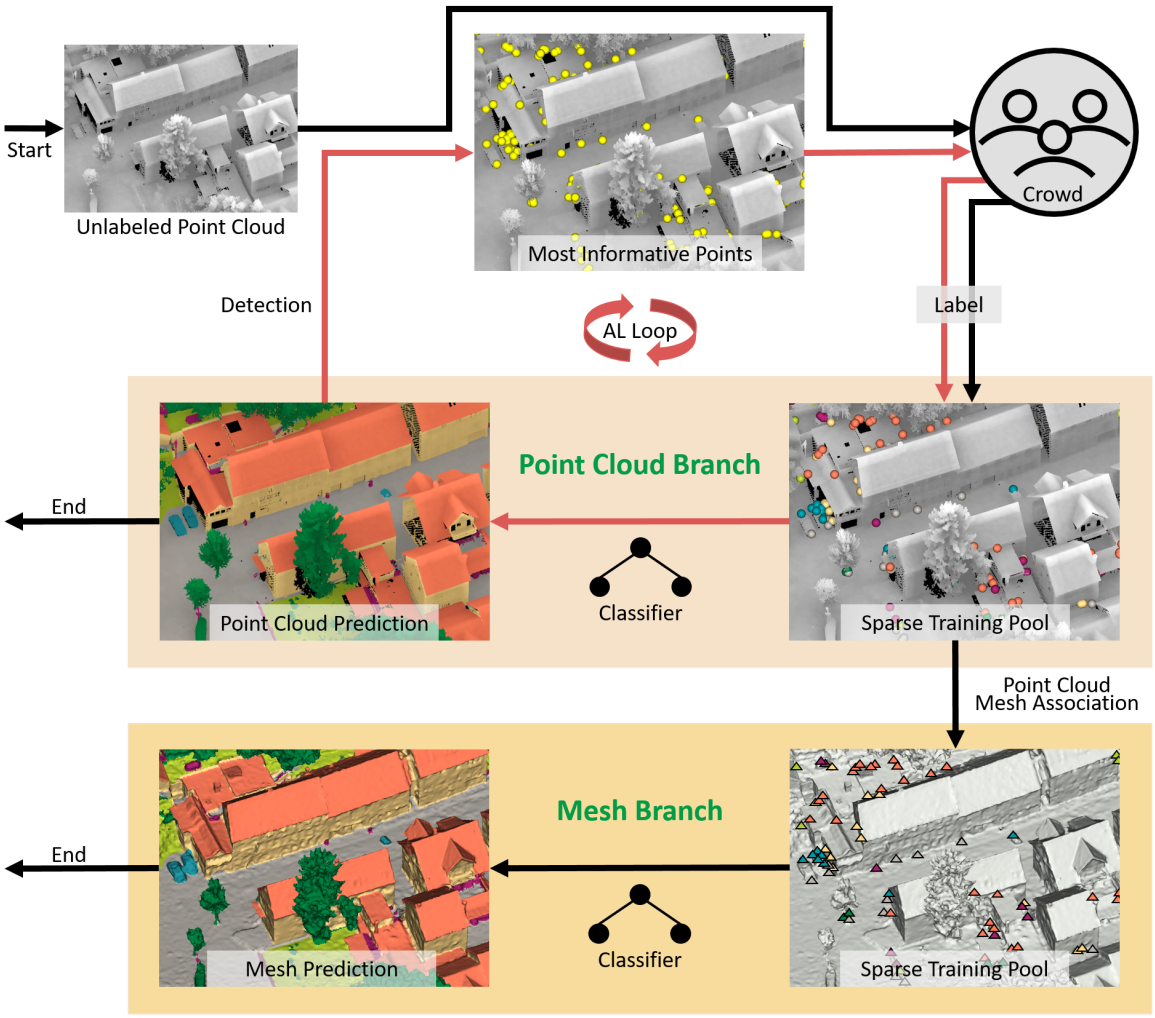


Figure 1. Crowd-based Active Learning pipeline for coupled semantic segmentation of both the point cloud and the mesh.

mesh (see section 2.3).

2.2 Employed Features for Semantic Segmentation

We distinguish between features which can be computed from ALS measurements solely and features only available after deriving a textured 3D mesh. For the first group, we compute geometric features as proposed by Weinmann et al. (2015) and Chehata et al. (2009) by estimating the structural tensor for the local neighborhood of each individual point. After extracting the eigenvalues of that tensor, we can determine the characteristics of the respective point distribution by forming different ratios of eigenvalues (Weinmann et al., 2015). Since computing eigenvalues and eigenvectors eventually means to fit a plane to the local point distribution, we can further enhance our feature vector by taking into account the orientation of this locally adapted plane. Furthermore, we consider height based features by determining the height above ground (i.e., DTM level) for each LiDAR point. Additionally to purely geometric features, LiDAR inherent features such as echo ratio and intensity of received echo are also used for the segmentation process. In order to establish a multi-scale approach and to analyze features on different levels of abstraction, we compute each feature for spherical neighborhoods of radii $r = 1, 2, 3$ and 5 m.

To derive mesh features, we follow the approach of Tutzauer et al. (2019) and encode each individual face by its CoG. By this, we can on one hand compute all aforementioned geometric features for our CoG cloud. On the other hand, we preserve features of the mesh geometry (approximated by CoGs) by assigning mesh inherent features such as area/density of faces, normal orientation and curvature, to the respective CoGs (i.e., faces). Furthermore, by using PCMA, LiDAR-specific features can be transferred to the mesh representation. For both the point cloud and the mesh, we additionally incorporate radiometric features (required that respective imagery is available). For this, RGB tuples are converted to HSV color space and used together with Gaussian smoothed color values for the aforementioned spatial neighborhoods.

2.3 Employment of Crowdworkers

Although the ML part of our hybrid intelligence system is capable of identifying most informative points in order to improve the performance of its predictions, it still depends on a human annotator it can learn from. However, when employing the crowd for labeling, we need to consider that non-experts work on this task. Therefore, we need to design such a task as easy understandable as possible. For reaching a vast pool of potential crowdworkers, we develop web-based labeling tools similar to those presented in Kölle et al. (2020) and employ our crowdworkers by usage of

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2021-93-2021 | © Author(s) 2021. CC BY 4.0 License.

Figure 2. Webtool used by crowdworkers for labeling a selected point. Either the colorized point cloud (*left*) or the textured mesh (*right*) is presented to the crowd (please note that class designations were simplified for crowdworkers).

the *Microworkers* platform. In order to initialize the AL loop by a small training dataset T, a first crowd campaign is conducted where each crowdworker is asked to mark one point for each class. We ask for the results of 100 crowdworkers, each receiving a payment of 0.10 \$. Labeled points obtained by this first group of crowdworkers are afterwards outsourced to be controlled by 3 independent workers (minimum configuration for majority vote). Furthermore, we can easily enhance this crowd task by including control points. Precisely, we additionally include 4 control points, where the true answer is known. By this, we can on one hand filter low quality results (more than one point falsely annotated causing denial of payment). On the other hand, we encourage the crowdworkers to work carefully by paying an additional bonus of 0.05 \$ per task if only correct answers are given. Afterwards, we aggregate votes of the controlling group and only consider the labels received in the first place to be true if the majority approves. Otherwise these points are rejected. For comparability reasons, this initialization dataset is used for all AL runs.

In the AL loop, points selected by the machine (i.e., the RF) are subdivided in jobs of 10 points each. Here, we again aggregate class votes from 3 crowdworkers per point and include control jobs. Analogous to the previous control campaign, we add 4 control points and handle payment same as before (note that we always use the same 4 control points for each iteration step of all AL runs in order to allow comparability). For labeling these points, crowdworkers use the tool visualized in Figure 2. Sampled points are indicated in yellow. For enabling interpretation of such points we also need to provide their vicinity. Therefore, we present to the crowd all entities within a 2.5D neighborhood radius of 20 m around the selected point. To determine which data representation form is easier for the crowd to understand, this neighborhood is i) given by the colorized 3D point cloud and ii) by the textured mesh and evaluated separately.

2.4 Dataset and Ground Truth Generation

2.4.1 Data Acquisition and Processing. For evaluating our proposed pipeline, we utilize simultaneously acquired LiDAR data and imagery captured over the village of Hessigheim, Germany (Haala et al., 2020; Cramer et al., 2018). Our setup is constituted of a *Riegl VUX-1LR* scanner and two oblique looking *Sony Alpha 6000* cameras, integrated on a *RIEGL Ricopter* platform. Considering a height above ground of 50 m, we achieve a laser footprint of less than 3 cm and a Ground Sampling Distance for the cameras of $1.5 - 3$ cm. Georeferencing of acquired LiDAR strips of this highly dense LiDAR point cloud (800 pts/m^2, approx. 126 M points in total) is accomplished using the *OPALS* software (Pfeifer et al., 2014). The LiDAR point cloud is furthermore colorized by nearest neighbor interpolation of colors from a photogrammetric point cloud derived from *Sony* imagery in order to make the data more familiar to crowdworkers. However, especially for vegetation where Dense Image Matching fails to generate accurate 3D point clouds due to lacking detection of identical points in image space, color information can only insufficiently be mapped to the LiDAR point cloud (see Figure 2).

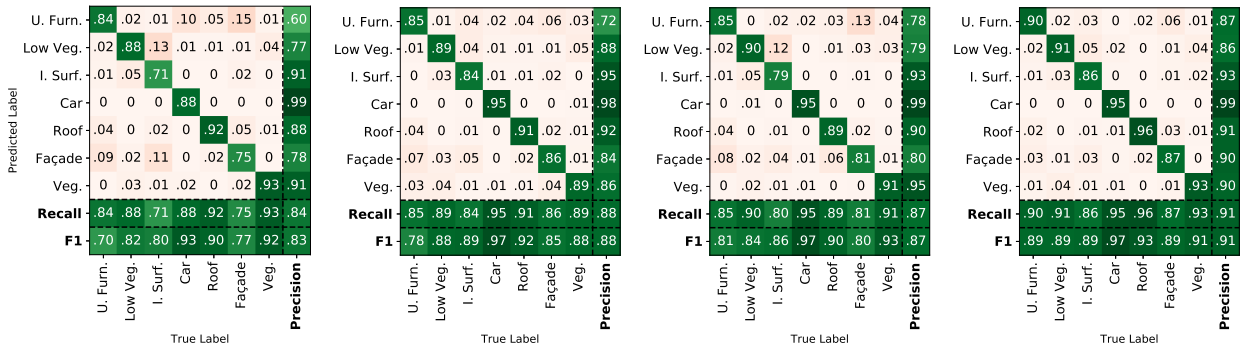
In case of the textured mesh, color information is directly mapped from the acquired images even considering occlusions by the help of the distinct 3D surface, given by the mesh geometry. For generating this second data representation, both the LiDAR point cloud and imagery are processed in a hybrid manner by combination of *OPALS* and *SURE* software (Pfeifer et al., 2014; Rothermel et al., 2012).

2.4.2 Creation of Ground Truth Data. Reference data for the point cloud was manually provided by the authors as outlined in Kölle et al. (2021). For our study, we decide to only use a subset of available classes (i.e., we merge classes), since in Kölle et al. (2020) it is observed that some classes are hard to interpret for non-experts. Precisely, we keep classes *Urban Furniture* (including *Powerline*), *Low Vegetation* (including *Gravel*), *Impervious Surface*, *Car*, *Roof* (including *Chimney*), *Façade* and *Vegetation* (*Shrub* & *Tree*). Ground Truth of the 3D mesh is obtained by transferring labels from the fully labeled point cloud to the mesh via PCMA.

3. RESULTS

Within this section, we first discuss the conducted experiments relying on real crowdworkers and some details regarding the crowd campaigns (section 3.1), which run in parallel to our AL loops. The performance of the AL loops (i.e., the performance of the machine learning from the crowd) is elaborated in section 3.2.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2021-93-2021 | © Author(s) 2021. CC BY 4.0 License.



(a) $\mathcal{O}_{CP}$ (b) $\mathcal{O}_{CP}$ + *RIU* (c) $\mathcal{O}_{CM}$ (d) $\mathcal{O}_{CM}$ + *RIU*

Figure 3. Achieved labeling accuracies of crowd oracles both with and without *RIU*, representing the vicinity of a point either as point cloud (*a* and *b*) or as 3D mesh (*c* and *d*).

For the point cloud we run 6 different AL loops:

- AL($\mathcal{O}_O$) and AL$_{RIU}$($\mathcal{O}_O$) denote the simulated AL loops using an omniscient oracle $\mathcal{O}_O$, always predicting the true label for the selected AL points, whereby AL$_{RIU}$($\mathcal{O}_O$) makes use of *RIU* in order to check whether increasing distance to decision boundary diminishes reachable accuracy.
- AL($\mathcal{O}_{CP}$) and AL$_{RIU}$($\mathcal{O}_{CP}$) refer to the same two AL approaches (with and without *RIU*) using a real crowd oracle where the point cloud is presented to crowdworkers ($\mathcal{O}_{CP}$).
- For AL($\mathcal{O}_{CM}$) and AL$_{RIU}$($\mathcal{O}_{CM}$) the point cloud is replaced by the 3D textured mesh.

For all experiments d_{RIU} is set to $1.5\,\mathrm{m}$ according to the findings of Kölle et al. (2021).

While in section 3.2.1 the performances when using crowd labels for semantic segmentation of the point cloud are reported, section 3.2.2 is dedicated to the evaluation of the results of the semantic mesh segmentation based on the same labels as for the respective point cloud runs but transferred to the mesh via PCMA.

As a compromise between total costs and segmentation accuracy, each run is conducted for $n_i = 10$ iteration steps, whereby in each step 300 points are selected for labeling. Since we aim on coupling the mesh branch with the point cloud branch, as a first preprocessing step, we eliminate all points in the point cloud which cannot clearly be assigned to an entity (i.e., face) of the 3D mesh. By this, we solely reduce the number of points in our unlabeled training pool U, but guarantee clear correspondences between point cloud and mesh. For efficiency reasons, we further apply a spatial subsampling to 30 cm point distance for the training point cloud, which again reduces the number of instances in U. This is however justified since this method can be considered as eliminating quasi duplicates in feature space. In case of our dense dataset, close neighbors of individual points incorporate very similar feature vectors. Since AL aims on selecting most informative points avoiding duplicates, our spatial subsampling not only boosts processing speed but also helps guaranteeing diversity of selected points (Kölle et al., 2021).

3.1 Performance of the Crowd

To evaluate which data representation is best suited for presenting to crowdworkers, Figure 3 displays the confusion matrices derived from the labels obtained by the crowd for a complete AL run each. First of all, we can observe that when spatial context is given by means of the textured mesh, Overall Accuracy (OA) is significantly higher (84 % vs. 87 %). This means that understanding meshes is significantly easier for non-experts (i.e., the crowd) substantiating our initial hypothesis. When we present the point cloud to the crowd for labeling ($\mathcal{O}_{CP}$), most confusion can be found regarding class *Urban Furniture*. Precisely, the crowd tends to label a point as *Urban Furniture* aka class *Other* (see Figure 2) whenever a difficult to interpret point is presented. Using such a class always poses the risk that crowdworkers select this class for most points in order to complete the task as fast as possible and receive their payment (Gadiraju et al., 2015). This underlines the importance of our control points, which allow the results of crowdworkers following this behavioral pattern to be excluded. Nevertheless, repeatedly choosing class *Other* might also be an indication that crowdworkers have severe difficulties in interpreting many points. In contrast, for meshes, this issue is already alleviated due to the more realistic closed 3D surface data representation (see Figure 2). Further confusion for $\mathcal{O}_{CP}$ can be observed for *Low Vegetation* vs. *Impervious Surface*, *Roof* vs. *Façade* and *Impervious Surface* vs. *Façade*. Those misclassifications can be explained due to the adjacent occurrence of these classes in real 3D scenes.

Since AL focuses on selecting points the classifier is most uncertain about, eventually points lying on class borders are sampled (Ertekin et al., 2007; Kölle et al., 2021). Such points are of course ambiguous for interpretation and might even be mixed by experts. This can also be observed for the labels received by $\mathcal{O}_{CM}$ relying on the mesh. Therefore, when increasing the distance to the decision boundary by *RIU*, both for presenting the point cloud or the mesh to the crowd, we can increase the labeling accuracy and especially minimize such misclassifications since now a clear class label can be assigned. In both cases *RIU* increases the OA by about 4 percentage points, which demonstrates a wise choice of query functions is profitable in AL, especially minding the employed oracle. With an OA of 91 % we can yield top label results by enhancing the query function by *RIU* and relying on the textured mesh as presentation modality (see Figure 3).

While accuracies reported in Figure 3 only indicate the performance of the respective oracles with regard to the complete loop, in Figure 4 (*top*) we oppose the OA of our individual runs at each iteration step to each other. Please note that we use OA instead of mean F1-score, since it would only poorly represent labeling accuracy, as in many iteration steps only a few points for some

classes are present (see also Figure 5). We can observe that labeling OA decreases in the course of the AL loop for all oracle types. Depending on the initial training set, points the classifier is most confused about might just be those of which no similar ones are included in the training set so far and which are not necessarily complex for interpretation. During the AL loop, points sampled become consequently more demanding for labeling since the more confident the model is about typical objects depicted in the point cloud, the more specific the selections of the model get - which in the end explains the continuously loss in OA. While the general ranking of our different oracles remains same as in Figure 3, $\mathcal{O}_{CM}$ + *RIU* shows an almost constant OA, further underlining its effectiveness.

In our experiments, the time required to complete the crowd jobs of one iteration step is less than 11 hours. This means that a complete AL run is finished in about 5 days (approx. 11 h · 10 iteration steps + approx. 16 h for initialization = 126 h).

3.2 Performance of the AL Loop

The results presented in section 3.1 are utilized within the AL loop for training of our RF models, which are in all experiments parametrized by 100 binary decision trees with maximum depth of 18 (empirically determined by grid search). In each iteration step, the models for the point cloud and the mesh are trained from scratch applying bootstrapping and afterwards used for inference on the test dataset of H3D. Results are reported in Figure 4 (*middle* & *bottom*) in terms of mean F1-score.

3.2.1 Semantic Segmentation of the Point Cloud. For the point cloud, in Figure 4 (*middle*) we can observe that within the first 4 iteration steps all approaches perform similarly well and almost reach accuracies of the baseline solutions, which utilize correct labels only (AL($\mathcal{O}_O$) & AL$_{RIU}$($\mathcal{O}_O$)). From the fourth iteration step on, the approaches relying on meshes as presentation modality (AL($\mathcal{O}_{CM}$) & AL$_{RIU}$($\mathcal{O}_{CM}$)) start diverging from the ones using point clouds (AL($\mathcal{O}_{CP}$) & AL$_{RIU}$($\mathcal{O}_{CP}$)). In the end of the iterations, AL runs where meshes are presented to the crowd perform up to 2.44 percentage points better in mean F1 and 2.71 percentage points in OA (see Table 1). While in case of $\mathcal{O}_{CM}$ not only the performance of the crowd but also the one of our RF seems to benefit from *RIU*, the latter does not apply to our AL loop depending on $\mathcal{O}_{CP}$ (AL$_{RIU}$($\mathcal{O}_{CP}$)). *RIU* neither seems to improve nor to diminish the accuracies, although more received crowd labels are correct and systematic confusions (as discussed in section 3.1) can be reduced. This together with the fact that AL($\mathcal{O}_{CM}$) performs significantly better than AL$_{RIU}$($\mathcal{O}_{CP}$), although having a similar OA in crowd labels, might seem counterintuitive. The explanation for this can be found by analyzing the number of correctly labeled instances of class *Urban Furniture*, which is as follows: AL($\mathcal{O}_{CP}$): 241; AL$_{RIU}$($\mathcal{O}_{CP}$): 244; AL($\mathcal{O}_{CM}$): 376; AL$_{RIU}$($\mathcal{O}_{CM}$): 413. This corresponds well to the result of our RF model. The more correct samples for this particular class are available in the training set, the better the overall performance of the RF. This also explains the similar accuracies of AL($\mathcal{O}_{CP}$) and AL$_{RIU}$($\mathcal{O}_{CP}$). However, one would assume that AL is capable of detecting required points and requesting corresponding labels by design. Since this is evidently not the case, it can be concluded that the model is rather confident about this class - or at least is more unconfident in other classes. This can also be seen in Figure 5. Until the fourth iteration step, in total for all AL runs a similar amount of *Urban Furniture* points is queried. From then on, for the runs relying

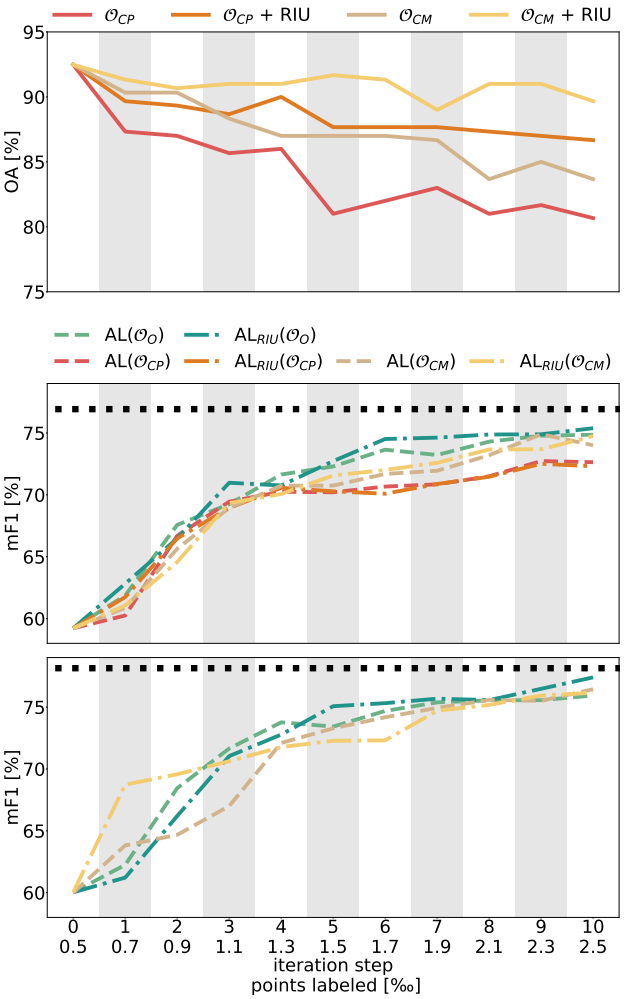


Figure 4. Comparison of the performance of the RF model regarding the **point cloud (*middle*)** and the **mesh (*bottom*)** for semantic segmentation, denoted as AL($\mathcal{O}$) and the accuracies of different **oracles** $\mathcal{O}$ **(*top*)** serving as input for the classifiers. *Dotted black* lines represent the baseline of each respective PL result. Please note that while for the AL runs mean F1-score is used for evaluation, for the reached accuracy of our crowd oracles we rely on OA (see also section 3.1).

on $\mathcal{O}_{CP}$ (AL($\mathcal{O}_{CP}$) & AL$_{RIU}$($\mathcal{O}_{CP}$)) almost no points are requested for this class, unlike AL loops using $\mathcal{O}_{CM}$ (AL($\mathcal{O}_{CM}$) & AL$_{RIU}$($\mathcal{O}_{CM}$)). This is mainly due to confusion by the crowd of points in class *Urban Furniture* with other classes (causing a low precision, see Figure 3) in early iteration steps, so that our model might become more unconfident about these other classes and therefore mainly requests labels of the remaining classes.

Table 1 further compares the reachable accuracy of PL (i.e., using the completely labeled training set) to our AL runs. For the point cloud, our best AL approach AL$_{RIU}$($\mathcal{O}_{CM}$) performs only 1.9 percentage points less in OA compared to PL and only 0.34 percentage points less compared to the baseline solution using the omniscient oracle $\mathcal{O}_O$. We want to stress that for this, we ask the crowd to label 0.25 % of available training points only, which causes costs of 190 \$ (100 points · 0.10 \$ + 100 points · 3 rep. · 0.15 \$ + n_i · (n/10) · 3 rep. · 0.15 \$, see section 2.3).

3.2.2 Semantic Segmentation of the 3D Mesh. For the coupled semantic segmentation of the 3D mesh (as depicted in Fig-

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2021-93-2021 | © Author(s) 2021. CC BY 4.0 License.

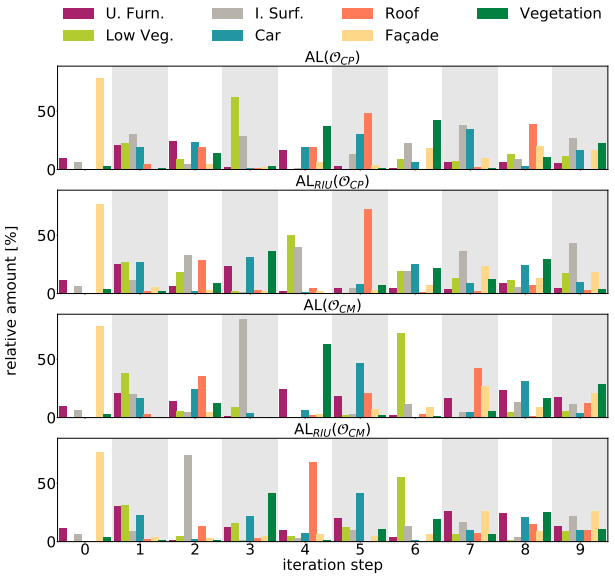


Figure 5. Histograms of true class affiliation of selected most informative points in each iteration step.

ure 1), AL runs for meshes perform similar to respective point cloud loops. Please note that AL runs depending on an oracle using point clouds ($\mathcal{O}_{CP}$) are not evaluated, since employing the intermediate result of a 3D point cloud is not reasonable when a derived mesh is already available. While our two baseline AL runs, where $\mathcal{O}_O$ is employed, yield top results in most iteration steps, the approaches using real crowd oracles (AL($\mathcal{O}_{CM}$) & AL$_{RIU}$($\mathcal{O}_{CM}$)) finally achieve similar accuracies (see also Table 1). However, in case of the segmentation of meshes, AL$_{RIU}$($\mathcal{O}_{CM}$) improves its mean F1-score significantly in the first iteration step. Afterwards the mean F1-score increases close to linearly while being clearly ahead of AL($\mathcal{O}_{CM}$) until the fourth iteration step. This behavior is desirable, since it offers the possibility of early stopping of the AL run, thereby further minimizing costs. From Table 1 we can observe that AL$_{RIU}$($\mathcal{O}_{CM}$) reaches top OA for depending on real crowdworkers and only differs by 2.5 percentage points from the baseline result of PL. Furthermore, in case of our 3D mesh, class *Urban Furniture* achieves a significantly higher F1-score compared to the point cloud. This can be explained by the fact that we only evaluate accuracies for the faces which can be clearly matched to labeled LiDAR data. Eventually this equals implicit filtering of class *Urban Furniture*, which is especially demanding in mesh generation due to rather complex and/or thin structures. Thus, only unambiguous representatives of this class are present in the mesh. Additionally, we want to stress that all point selections of the AL loop are determined by the point cloud classifier (as mentioned in section 2.2), which means that the detected most informative points might not be optimal for the mesh RF model. Nevertheless, by coupling the mesh branch labeling effort and thus costs are significantly reduced and accuracies similar to those from the non-approximated point cloud branch can be achieved (see Table 1).

4. CONCLUSION AND OUTLOOK

Within this work, we have proposed a human-in-the-loop pipeline, which is capable of completely avoiding the involvement of an expert in the tedious and costly labeling process of 3D geodata for classification purposes. By leveraging AL capabilities, we significantly reduce the amount of necessary labels, so that only 0.25 % of available training points require annotation by a human operator (which is in our case represented by the crowd), thus as well minimizing costs to 190 $. We also focus on investigating which form of data representation is best suited for presenting to crowdworkers and which modifications of the AL selection strategy help non-experts to label individual instances. We found that using meshes can improve labeling accuracy by about 3 percentage points. Moreover, we can ease interpretability by *RIU*, which increases annotation accuracy by another 4 percentage points. Both enhancements directly impact the performance of our coupled classifiers. We achieve an OA differing from the PL approach by only 1.9 percentage points for the point cloud and 2.5 percentage points for the mesh. However, when relying on crowdworkers, the variety of classes to be used is limited since applying an excessive class catalog also raises the complexity of respective crowd tasks and would likely cause the crowd to be overwhelmed (Kölle et al., 2020). All experiments within this work were carried out for the ultra-high resolution H3D dataset, which means that we rely on close to optimal data input. Similar datasets are not yet available on a nationwide scale. Therefore, in our future work, we will verify our proposed methods on current state-of-the-art datasets having much lower point densities.

			F1-score [%]								
Method	Data	Oracle	*U. Furn.*	*Low Veg.*	*I. Surf.*	*Car*	*Roof*	*Façade*	*Veg.*	mF1[%]	OA[%]
PL	Point Cloud	-	41.14	90.68	85.26	51.63	92.96	83.77	93.05	76.92	88.16
	Mesh	-	48.57	91.38	88.15	49.81	91.86	85.34	91.88	78.14	88.00
AL	Point Cloud	$\mathcal{O}_O$	33.93	**90.31**	82.70	56.34	88.33	**79.73**	**92.66**	74.86	**86.65**
		$\mathcal{O}_O$ + *RIU*	**36.97**	89.91	**83.84**	55.42	**90.05**	79.61	91.91	**75.39**	86.60
		$\mathcal{O}_{CP}$	32.32	89.92	76.26	53.95	88.88	75.43	91.70	72.64	83.38
		$\mathcal{O}_{CP}$ + *RIU*	31.00	88.54	79.69	53.04	86.82	76.43	90.67	72.31	83.55
		$\mathcal{O}_{CM}$	33.37	88.34	78.14	**57.40**	88.89	79.83	92.07	74.01	85.15
		$\mathcal{O}_{CM}$ + *RIU*	34.56	89.59	81.81	56.20	89.18	79.35	92.56	74.75	86.26
	Mesh	$\mathcal{O}_O$	43.32	90.00	84.00	54.65	86.65	81.95	90.95	75.93	85.32
		$\mathcal{O}_O$ + *RIU*	**46.57**	**90.14**	**85.95**	57.37	**88.88**	81.91	90.99	**77.40**	**85.95**
		$\mathcal{O}_{CM}$	45.93	88.71	82.83	**57.51**	87.45	**82.08**	90.58	76.44	84.99
		$\mathcal{O}_{CM}$ + *RIU*	44.91	90.02	85.39	53.36	86.84	81.40	**91.15**	76.15	85.54

Table 1. Comparison of final classification results on the test site of H3D both for the ALS point cloud and the 3D mesh for PL and AL. The AL designations are related to our different oracle types.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2021-93-2021 | © Author(s) 2021. CC BY 4.0 License.

For those, we assume the mesh to outperform the point cloud as presentation modality by even greater margin.

ACKNOWLEDGEMENTS

The H3D dataset has been captured in context of an ongoing research project funded by the German Federal Institute of Hydrology (BfG). We would like to thank Ivan Shiller for his support in carrying out the crowd campaigns.

REFERENCES

Badrinarayanan, V., Kendall, A. and Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, pp. 2481–2495.

Breiman, L., 2001. Random Forests. *Mach. Learn.* 45(1), pp. 5–32.

Budhathoki, N. R. and Haythornthwaite, C., 2012. Motivation for Open Collaboration: Crowd and Community Models and the Case of OpenStreetMap. *American Behavioral Scientist* 57(5), pp. 548–575.

Buhrmester, M., Kwang, T. and Gosling, S. D., 2011. Amazon's Mechanical Turk: A New Source of Inexpensive,Yet High-Quality, Data? *Perspectives on Psychological Science* 6(1), pp. 3–5.

Chehata, N., Guo, L. and Mallet, C., 2009. Airborne LiDAR Feature Selection For Urban Classification Using Random Forests. *ISPRS Archives* XXXVIII-3/W8, pp. 207–212.

Cramer, M., Haala, N., Laupheimer, D., Mandlburger, G. and Havel, P., 2018. Ultra-High Precision UAV-Based LiDAR and Dense Image Matching. *ISPRS Archives* XLII-1, pp. 115–120.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Li, F. F., 2009. ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR 2009*, pp. 248–255.

Ertekin, S., Huang, J., Bottou, L. and Giles, L., 2007. Learning on the Border: Active Learning in Imbalanced Data Classification. In: *CIKM 2007*, ACM, New York, NY, USA, pp. 127–136.

Gadiraju, U., Kawase, R., Siehndel, P. and Fetahu, B., 2015. Breaking Bad: Understanding Behavior of Crowd Workers in Categorization Microtasks. In: *HT 2015*, ACM, pp. 33–38.

Glira, P., Pfeifer, N. and Mandlburger, G., 2019. Hybrid orientation of airborne lidar point clouds and aerial images. *ISPRS Annals* IV-2/W5, pp. 567–574.

Graham, B., Engelcke, M. and v. d. Maaten, L., 2018. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In: *CVPR 2018*, pp. 9224–9232.

Haala, N., Kölle, M., Cramer, M., Laupheimer, D., Mandlburger, G. and Glira, P., 2020. Hybrid Georeferencing, Enhancement and Classification of Ultra-High Resolution UAV LiDAR and Image Point Clouds for Monitoring Applications. *ISPRS Annals* V-2-2020, pp. 727–734.

Herfort, B., Höfle, B. and Klonner, C., 2018. 3D micro-mapping: Towards assessing the quality of crowdsourcing to support 3D point cloud analysis. *ISPRS Journal of Photogrammetry and Remote Sensing* 137, pp. 73–83.

Hirth, M., Hoßfeld, T. and Tran-Gia, P., 2011. Anatomy of a Crowdsourcing Platform - Using the Example of Microworkers.com. In: *IMIS 2011*, IEEE Computer Society, Washington, DC, USA, pp. 322–329.

Kölle, M., Walter, V., Schmohl, S. and Soergel, U., 2020. Hybrid Acquisition of High Quality Training Data for Semantic Segmentation of 3D Point Clouds using Crowd-Based Active Learning. *ISPRS Annals* V-2-2020, pp. 501–508.

Kölle, M., Walter, V., Schmohl, S. and Soergel, U., 2021. Remembering both the machine and the crowd when sampling points: Active learning for semantic segmentation of als point clouds. In: *ICPR International Workshops and Challenges*, Springer International Publishing, Cham, pp. 505–520.

Laupheimer, D., Shams Eddin, M. H. and Haala, N., 2020. On The Association of LiDAR Point Clouds and Textured Meshes for Multi-Modal Semantic Segmentation. *ISPRS Annals* V-2-2020, pp. 509–516.

Li, N. and Pfeifer, N., 2019. Active Learning to Extend Training Data for Large Area Airborne LiDAR Classification. *ISPRS Archives* XLII-2/W13, pp. 1033–1037.

Lin, Y., Vosselman, G., Cao, Y. and Yang, M. Y., 2020. Active and incremental learning for semantic ALS point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing* 169, pp. 73–92.

Lloyd, S. P., 1982. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* 28(2), pp. 129–137.

Luo, H., Wang, C., Wen, C., Chen, Z., Zai, D., Yu, Y. and Li, J., 2018. Semantic labeling of mobile LiDAR point clouds via active learning and higher order MRF. *TGRS* 56(7), pp. 3631–3644.

Pfeifer, N., Mandlburger, G., Otepka, J. and Karel, W., 2014. OPALS – a framework for airborne laser scanning data analysis. *Computers, Environment and Urban Systems* 45, pp. 125–136.

Qi, C. R., Yi, L., Su, H. and Guibas, L. J., 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: *NIPS 2017*, NIPS'17, Curran Associates Inc., USA, pp. 5105–5114.

Ronneberger, O., Fischer, P. and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: N. Navab, J. Hornegger, W. M. Wells and A. F. Frangi (eds), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Springer International Publishing, Cham, pp. 234–241.

Rothermel, M., Wenzel, K., Fritsch, D. and Haala, N., 2012. SURE: Photogrammetric Surface Reconstruction From Imagery. In: *Proceedings LC3D Workshop*, Vol. 8, Berlin.

Rouhani, M., Lafarge, F. and Alliez, P., 2017. Semantic Segmentation of 3D Textured Meshes for Urban Scene Analysis. *ISPRS Journal of Photogrammetry and Remote Sensing* 123, pp. 124 – 139.

Settles, B., 2009. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Tutzauer, P., Laupheimer, D. and Haala, N., 2019. Semantic Urban Mesh Enhancement Utilizing A Hybrid Model. *ISPRS Annals* IV-2/W7, pp. 175–182.

Vaughan, J. W., 2018. Making Better Use of the Crowd: How Crowdsourcing Can Advance Machine Learning Research. *Journ. Mach. Learn. Res.* 18(193), pp. 1–46.

Walter, V. and Soergel, U., 2018. Implementation, Results, and Problems of Paid Crowd-Based Geospatial Data Collection. *PFG* 86, pp. 187–197.

Weinmann, M., Jutzi, B., Hinz, S. and Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal* 105, pp. 286–304.

Zhdanov, F., 2019. Diverse mini-batch Active Learning. *CoRR*.