HYBRID ACQUISITION OF HIGH QUALITY TRAINING DATA FOR SEMANTIC SEGMENTATION OF 3D POINT CLOUDS USING CROWD-BASED ACTIVE LEARNING

M. Kölle[1,*], V. Walter[1], S. Schmohl[1], U. Soergel[1]

[1] Institute for Photogrammetry, University of Stuttgart, Germany -
(michael.koelle, volker.walter, stefan.schmohl, uwe.soergel)@ifp.uni-stuttgart.de

Commission II, WG II/6

KEY WORDS: Crowdsourcing, Active Learning, 3D Point Clouds, Labeling, Classification, Random Forest, Sparse 3D CNN, ISPRS 3D Semantic Labeling

ABSTRACT:

Automated semantic interpretation of 3D point clouds is crucial for many tasks in the domain of geospatial data analysis. For this purpose, labeled training data is required, which has often to be provided manually by experts. One approach to minimize effort in terms of costs of human interaction is Active Learning (AL). The aim is to process only the subset of an unlabeled dataset that is particularly helpful with respect to class separation. Here a machine identifies informative instances which are then labeled by humans, thereby increasing the performance of the machine. In order to completely avoid involvement of an expert, this time-consuming annotation can be resolved via crowdsourcing. Therefore, we propose an approach combining AL with paid crowdsourcing. Although incorporating human interaction, our method can run fully automatically, so that only an unlabeled dataset and a fixed financial budget for the payment of the crowdworkers need to be provided. We conduct multiple iteration steps of the AL process on the *ISPRS Vaihingen 3D Semantic Labeling* benchmark dataset (V3D) and especially evaluate the performance of the crowd when labeling 3D points. We prove our concept by using labels derived from our crowd-based AL method for classifying the test dataset. The analysis outlines that by labeling only $0.4\,\%$ of the training dataset by the crowd and spending less than $145\,\$$, both our trained Random Forest and sparse 3D CNN classifier differ in Overall Accuracy by less than 3 percentage points compared to the same classifiers trained on the complete V3D training set.

1. INTRODUCTION

In recent years Machine Learning (ML) techniques, especially Convolutional Neural Networks (CNNs), have gained more and more importance in the field of geospatial analysis of 3D point clouds. Whether for automated semantic segmentation (Weinmann et al., 2015; Niemeyer et al., 2014; Qi et al., 2017) or object detection in 3D point clouds (Feng et al., 2019; Yang et al., 2018) ML techniques can be applied. Reasonable applications can be found in filtering ground points for Digital Terrain Model generation (Hu and Yuan, 2016), reconstruction of 3D city models and surrounding awareness in context of autonomous driving. The training of such networks, however, usually requires enormous amounts of annotated data, to which semantic meaning has been assigned in the form of a so-called label. However, in the case of geodata, especially in three-dimensional space, only a few such data sets are available. In order to be able to use corresponding classification approaches efficiently also in the domain of the automatic interpretation of geodata, it is necessary to optimize the labeling process for setting up the training data set for this special data.

This can be achieved by Active Learning (AL) (Kovashka et al., 2016; Mackowiak et al., 2018). In Wortman Vaughan (2018), the AL process is referred to as a hybrid intelligence system where a machine interactively works with a human to combine the strengths of both parties. The aim is to process only the subset of the previously unlabeled data that is particularly helpful in terms of better class separation. Using a ML algorithm, the machine identifies informative instances, which are afterwards being annotated by a so-called oracle and added to the training pool. So, in a hybrid sense, the machine relies on input of a human annotator, thereby increasing its own performance. Therefore, such "human-in-the-loop" systems (Branson et al., 2010) are an efficient means to significantly reduce the label effort as recently shown in Kellenberger et al. (2019). Nevertheless, labels have to be provided manually, which is often done by experts. Since the labeling process is a very tedious and therefore costly task, there is an increasing interest in solving such simple label tasks by crowdsourcing.

Crowdsourcing comprises the idea of outsourcing tasks from a small group of specialists to an unknown large group of workers (Hetmank, 2013). However, incentives need to be offered to the crowdworkers. While they may be motivated by fascination or fun at work in case of voluntary contribution, the incentive can as well be given by payment. One prominent example in the context of Volunteered Geographic Information (VGI) is Open Street Map (OSM), where the incentive is given by collaboratively building up a publicly available product (Budhathoki and Haythornthwaite, 2012). In case of paid crowdsourcing, platforms such as *Amazon Mechanical Turk* (Buhrmester et al., 2011) can be used in order to offer a job to the crowd and manage payment, boni etc. However, the tasks need to be easy enough so that non-experts are capable of resolving it. In the context of establishing the *ImageNet* dataset (Deng et al., 2009) it was already shown that crowdsourcing can be very well applied for labeling RGB-imagery showing everyday life scenes. In detail, every image preselected via search engines was presented to multiple crowdworkers in order to check whether the image is labeled correctly by usage of majority vote.

*Corresponding author

In case of geospatial data, we are dealing with scenes rather unknown to common crowdworkers such as aerial images, where an unfamiliar perspective is inherent. Walter and Soergel (2018) analyzed the performance of the crowd when digitizing objects from aerial images in specific classes such as streets, buildings and forests. However, engaging crowdworkers for 3D data, places even higher demands on the crowd, since the interpretation of 3D data viewed on a 2-dimensional screen requires a distinct spatial imagination. Herfort et al. (2018) investigated the applicability of volunteered crowdsourcing for 3D point clouds by asking the crowd whether a given subset of a point cloud around a possible tree position contains noise in terms of points of other objects and which kind of objects. Furthermore they asked the crowdworkers to determine the height of the trunk of the trees by moving a plane to the top of the trunk. To the best of our knowledge this work is the only one investigating crowdsourcing for 3D point cloud analysis so far.

In this paper, we propose a paid crowd-based AL approach for efficient labeling of most informative points of a 3D point cloud from Airborne Laserscanning (ALS) data, which are afterwards used to train a classifier. This approach comprises a total of three different kinds of crowd jobs using several custom web-based tools as well each requiring a different level of skill. These jobs are important for initializing and conducting the AL pipeline by (1) requesting the crowd to pick sample points for each class, (2) control these samples by other independent crowdworkers and (3) asking the crowd for labels of specific complex-to-interpret points inhering most information in context of the AL procedure.

2. METHODOLOGY

In this section, we present our approach for establishing a highly informative training pool from a given ALS point cloud (section 2.2) in a fully automated process chain (section 2.1). This method consists of three sequential types of tasks outsourced to the crowd each requiring a suitable web-based tool, so that crowdworkers can easily accomplish them (section 2.3). Furthermore, these tasks need to be assigned to the crowd using a proper platform for hiring and managing crowd jobs (section 2.4).

2.1 Crowd-based AL Pipeline

The basic idea of this work is based on the principle of AL (Settles, 2009), which aims at reducing the training data annotation effort. In contrast to passive learning, where an already labeled data set is provided to the classifier, here the interaction of the classifier with an operator during the labeling process is in focus. Specifically, the classifier actively selects points to be added to the training data set due to their high information content, starting from a initial training data set of small size. Since these selected points have to be processed manually, the machine is dependent on the input of an operator. In order to not only reduce the costly deployment of an expert, but to avoid it altogether, the approach of a purely crowd-based acquisition process is pursued in this work. The special feature is that, although human interaction is necessary in this process, it can be fully automated, since informative instances are automatically queried using a ML algorithm. From perspective of an employer, labeling these instances by a crowdworker can be compared with calling a function within a program. Posting a task on a crowdsourcing platform, retrieving results and paying crowdworkers can be accomplished using a programming interface (API). This allows realizing fully automated processes where parts of the task are performed by humans. Thus an employer only has to provide a point cloud from which the training data set is to be extracted, as well as the budget for the remuneration of the crowdworkers. The basic workflow is visualized in Figure 1.

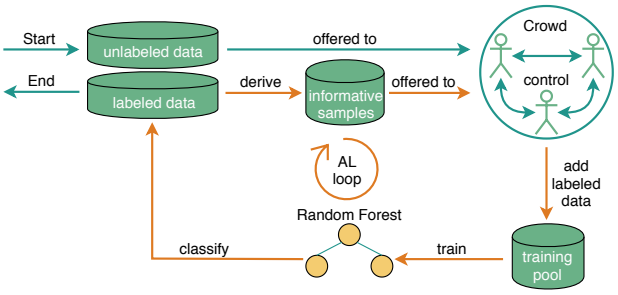


Figure 1. Crowd-based Active Learning pipeline.

In order to provide the previously mentioned first training data set, the first task of each crowdworker is to label a representative point for each defined class (see Figure 2 (a)) from a given point cloud. Yet the obtained labels, typically have a rather low information content. This is due to the fact that informative points in feature space as well as in object space are primarily located at the decision boundaries (Ertekin et al., 2007). However, paid crowdworkers generally are prone to maximize their income, maybe even by means of dishonesty (Wortman Vaughan, 2018), while minimizing the required time. Therefore, crowdworkers typically tend to select points further away from class boundaries because they can be easily and quickly assigned to a class.

However, it has to be expected that points may be labeled falsely. Either the crowdworker does not manage to find a point of the respective class or simply wants to quit the task as soon as possible. These misclassifications are quite harmful to the performance of the upcoming classifier. Because the crowdworkers are free to decide which point to provide for a class, it is likely that no point will be used by more than one crowdworker. Therefore, in the context of this task no reasonable controlling mechanism can be applied. As a remedy each labeled point of the first stage of initializing will be presented to another crowdworker, so that he is controlling the result of the first crowdworker. This task itself is quite easy to accomplish since the outcome is only a binary decision. Especially when proposing easy and maybe repeating tasks to the crowd the danger of crowdworkers choosing answers randomly in order to speed their operation time rises (Gadiraju et al., 2015). Therefore controlling mechanisms in the form of control samples need to be realized as well, so that malicious crowdworkers can be identified and hence their control job does not influence the pipeline. After the initialization, we assume to have a pool of correctly classified samples for each class, so that a first training pool containing all manually labeled points can be established.

As next step, this training pool is used to train a classifier. For the sake of clearness, for training, we provide the full point cloud to the classifier so that we can derive meaningful features based on point neighborhoods for our few labeled points. The task of the classifier is twofold. First, it predicts labels for each remaining unlabeled point, whereby a first complete classification is given, however of still low quality. Second, and that is more important to our purpose, based on this class prediction the classifier identifies those points that can only be classified with low reliability so far, whether because such a point was not part of the first training set or it is a point near the decision boundary. These are precisely

those points with a high information content that are to be added to the training pool.

The design of the query function for detecting such points is key in order to increase the performance of the model by building up a diverse training set. According to Settles (2009) the most common selection strategies are *uncertainty sampling* and *query-by-committee*. While *uncertainty sampling* is based on the a posteriori probability $P(c|x)$ that point x belongs to class c measured most commonly in terms of entropy, in *query-by-committee* a pool E of e distinct models in an ensemble classifier can be used for measuring informativeness as disagreement among different models. In this context Argamon-Engelson and Dagan (1999) introduced Vote Entropy VE (equation 1).

$$VE = -\sum_{c} \frac{\sum_e D(\mathbf{P}_e, c)}{N_E} \cdot \log \frac{\sum_e D(\mathbf{P}_e, c)}{N_E}$$
$$\text{where } D(\mathbf{P}_e, c) = \begin{cases} 1, & \text{if } \operatorname{argmax}(\mathbf{P}_e) = c \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Each member e of the ensemble E predicts an a posteriori probability for each class c which is stored in $\mathbf{P}_e$. Based on $\mathbf{P}_e$ every committee member favors one specific class which is seen as the vote of this model. All votes are afterwards summed per class and then normalized by the number of the ensemble members N_E and inserted into Shannon's entropy formula (if no ensemble member votes for a specific class, i.e. $\sum_e D(\mathbf{P}_e, c) = 0$, VE is not increased). The main advantage of VE over averaged entropy of committee members is, that points having low maximum a posteriori probabilities, are not necessarily selected as long as every member is still voting for the same class. In order to provide such an ensemble classifier we rely on the Random Forest (RF) (Breiman, 2001). For this RF classifier we are using both geometric and radiometric features as described in Kölle et al. (2019) and Haala et al. (2020).

This combination of RF and query function establishes an efficient automated process for sampling n valuable points for manual labeling in contrast to randomly sample points. Based on the selection strategy either one instance having the greatest VE in each iteration step (*stream-based* AL) or a small pool containing primitives having the n top VE-values (*pool-based* AL) can be selected. Since *stream-based* AL is not efficient in the context of our approach, the top n points with highest VE will be proposed again to the crowd as third type of crowd task. Here, the crowdworkers are asked to label exactly these shown points (see Figure 1) which may be more complex for crowdworkers to label since often such points are lying on the border between two classes both in feature space and 3D object space. Again, this task can be controlled easily by including control jobs.

After labeling these highly informative samples by the crowd, the points are added to the training pool in order to re-train the RF for classifying the remaining data. This process is repeated iteratively so that the training data set is built up step by step from points with a high information content, with the aim of gradually increasing the performance of the classifier. The iteration is continued until the result corresponds to the desired quality or the label budget is exhausted. An interpretation of this concept is a teacher showing his student how to perform tasks, whereas the student asks for help whenever difficult, new tasks appear. The student in AL is the machine and the ML algorithm respectively. The teacher, the oracle, is a human annotator, who in our case is represented by the crowd.

2.2 Dataset

For all our experiments we are using the *ISPRS Vaihingen 3D Semantic Labeling* (V3D) dataset (Niemeyer et al., 2014) containing suburban area typical for western world countries. The point density is about $4 - 8\,\mathrm{pts/m^2}$. In order to increase familiarity of the point cloud data to the crowdworkers we are colorizing the points by orthogonal projection of an orthophoto received from the author of Cramer (2010). One problem of this procedure is, that shadows are mapped onto the points and of course all points occluded by others are colored suboptimally by the occluding color. Regarding dynamic objects such as cars an additional issue occurs due to deviant acquisition times of the LiDAR data (Aug 21st, 2008) and the imagery (Aug 6th, 2008).

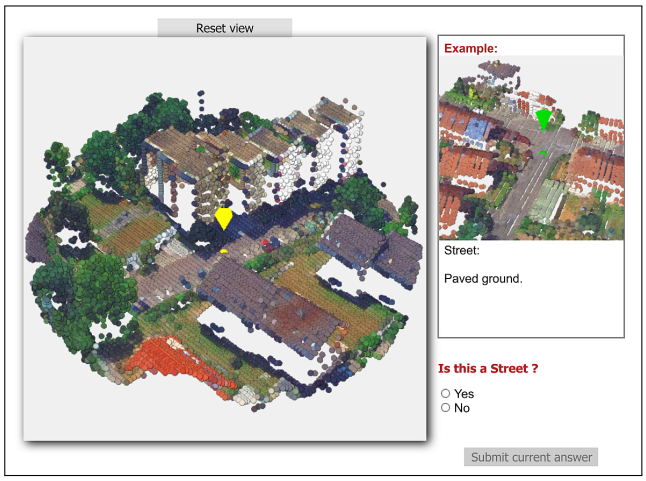
2.3 Task Design

This section briefly discusses the implementation of our three types of crowd-based acquisition tools and what the crowdworkers are asked for. Since we aim to reach a vast pool of motivated crowdworkers, such tools need to be made accessible via the internet. These web-based tools are implemented in HTML and Javascript based on the *three.js library* (Cabello, 2019) for basic website layout and interactive functionality respectively. These websites are hosted on our server relying on PHP for server communication such as distributing the correct dataset to the crowdworker and managing the submitted results.

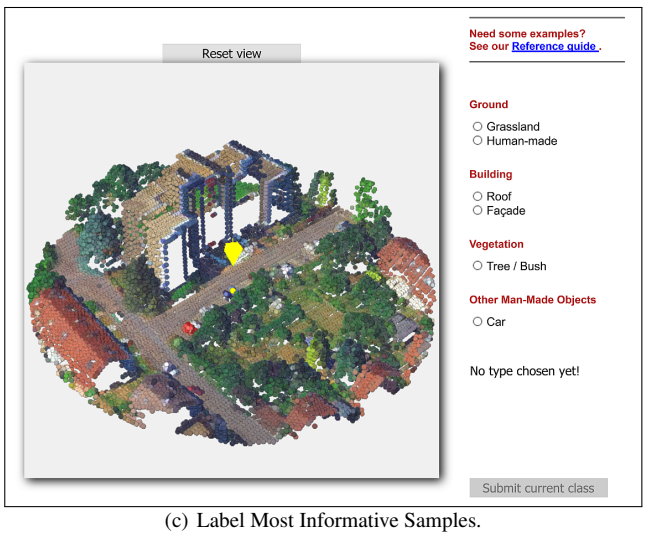
2.3.1 Type A: Point Picking and Labeling. As mentioned in section 2.1, for initializing the AL process, every crowdworker is asked to mark one point for each class (see Figure 2 (a)). In order to ease navigation in the point clouds and minimize loading time, the complete point cloud is split in multiple subsets, each offered to the crowd. The main window on the left side is dedicated to the visualization of the given point cloud subset, while on the right side controls are situated. For better navigation, whenever the cursor hovers over the point cloud, it is rotating slowly around z-axis but at any time the crowdworker is able to navigate on his own by rotating, panning or zooming. Except for zooming, navigation is only allowed while in viewing mode. As soon as the crowdworker enters picking mode the point cloud is fixed so that a point corresponding to the checked button can be selected without accidental movement. All selected points are indicated by an arrow colored same as the respective button. If the crowdworker has nevertheless selected a point by accident or wants to change his selection this point can as well be removed. As soon as one representative of each class has been selected, the crowdworker is allowed to submit the task.

2.3.2 Type B: Control Labeling. The second web tool is designed for controlling the points proposed by the first group of crowdworkers. Therefore as shown on the left of Figure 2 (b), a selected and labeled point is displayed including its surrounding points within a radius of 50 m in order to preserve context which is essential for interpretation. The selected point itself is indicated in yellow and further highlighted by a yellow arrow. The crowdworker is asked whether the highlighted point belongs to the class stated by a worker of the first stage. In order to assist the crowdworker, a typical representative of the respective class is given including a brief description. Since every worker controls the labeling job of another one of the first stage, this job consists of a total of nine points (one for each class). For evaluating the performance of every crowdworker, we further include control jobs. For this we provide three points, one labeled correctly, two falsely which are evenly distributed in every crowd

(a) Point Picking and Labeling.



(b) Control Labeling.



(c) Label Most Informative Samples.

Figure 2. Web-based tools used by crowdworkers. Every website further contains brief instructions and an example.

job. Furthermore the first point to be checked is a control point which is repeated as last point in order to check consistency of the answers. Therefore, every job requires 13 answers.

2.3.3 Type C: Label Most Informative Samples.

The third web tool we employed, is dedicated for labeling most informative samples selected by our classifier in each iteration step of the AL procedure. As seen in Figure 2 (c), the basic layout of this tool is according to Type B, whereas crowdworkers are asked to select the class of nine shown points by clicking the corresponding button. Again, control points are included, distributed the same as for Type B. Each crowdworker is controlled by the same three rather easy to label points which need to be provided by the employer.

2.4 Employing the Crowd

In order to hire crowdworkers we are using the *Microworkers* platform which has been analyzed in detail in Hirth et al. (2011). According to Weblabcenter Inc. (2019) the platform provides access to an international workforce of about 1.4 million workers who are categorized in different groups such as *Top Performers*, *All EU Workers*, *Top EU Workers* etc. As soon as a task is posted and the resulting fees are transmitted to *Microworkers*, the task is active and all crowdworkers of the specified group are allowed to work on this task. Within the instructions on *Microworkers* we provide the URL to our web tool hosted on our server. After completion of the task, crowdworkers receive a proof code generated on our website which is required for receiving payment. By using *Microworkers* we are able to conduct our experiments based on real crowdworkers we did not train or instruct. Therefore the results of all experiments, although of course depending on the participating crowdworkers, can be considered as generalized.

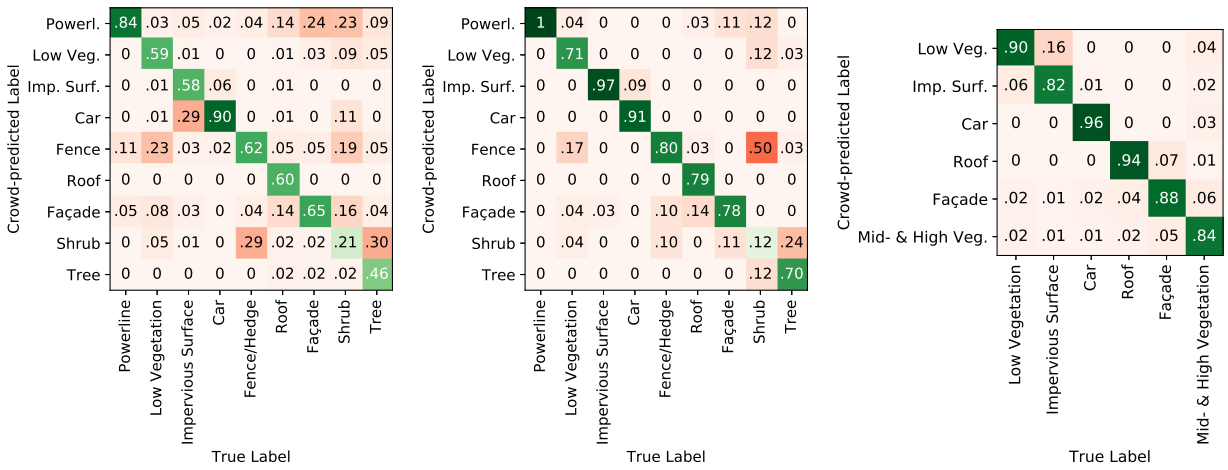
3. RESULTS

Within this section our results of applying the AL pipeline on the V3D dataset are presented. Although, within our AL pipeline data collection and learning from this data are incorporated, we first evaluate the labels provided by the crowd in section 3.1, whereas in section 3.3 the results of the AL pipeline are discussed which are however depending on the performance of the crowd.

3.1 Performance of the Crowd

We conducted a total of ten crowd campaigns (2 for initialization and 8 AL iteration steps) comprising 1016 (100 for picking, 100 for controlling and 102 per iteration step) tasks of labeling respectively controlling of 3348 ($100 \cdot 9$ classes $+$ batch size of $306 \cdot 8$ iteration steps) 3D points. Every task was assigned to the *Top Performers* group of *Microworkers* consisting of more than 2000 active workers.

In the context of picking points for each class freely (section 2.3.1), the crowd achieved a quite low Overall Accuracy (OA) of 56.44 %, what is to be expected since in this first step labels are directly given by single crowdworkers without any Ground Truth (GT) inference or controlling mechanisms. However, this crowd job serves as a first indicator for the capabilities of the crowd to distinguish different classes in 3D point clouds. The normalized confusion matrix is visualized in Figure 3 (a). In total 900 points have been labeled (100 tasks each asking for one point for each class). The matrix shows that points of other classes have often been erroneously labeled as *Powerline*. Since this class is naturally quite rarely represented in ALS point clouds, crowdworkers may not have managed to find a true point or at least not in a short time and therefore just selected randomly in order to finish the task. It can also be observed that crowdworkers have difficulties



(a) Point Picking and Labeling (b) Control Labeling (c) Label Most Informative Samples

Figure 3. Normalized confusion matrices for all three types of conducted crowd jobs.

distinguishing between vegetation classes such as *Fence/Hedge*, *Shrub* and *Tree*. While differentiation between a small tree and a shrub is already demanding for experts, it is even more complicated for crowdworkers because most of them are biased by typical vegetation in their home countries which can be quite different to western vegetation types. For example a fence surrounding an estate may be unknown to them and therefore confused with *Shrub*. Confusion between *Car* and *Impervious Surface* is due to the fact that the timestamps of acquiring the point cloud and the imagery used for colorization are not identical. Therefore, an unheedful crowdworker might label impervious surface as *Car* since plane street is colored as a car.

As a successive step, we carried out a controlling crowd job in order to improve the overall labeling accuracy. As already mentioned in section 2.3.2 control points were included in order to check how well the individual crowdworker performs. In this context we define four quality categories for the workers. *Category* 4 means the control label occurring twice as consistency test is answered same and correct. In *Category* 3 and 2, *Category* 4 is fulfilled and one more control point is passed whereby the control sample for *Category* 2 is more complex. In *Category* 1 all control jobs are passed. The allocation of the crowdworkers to the categories is delineated in Table 1, where number of workers is defined as all workers fulfilling at least minimum requirements of this category.

Category	# of workers	Control OA [%]	Label OA [%]
4	87	71.39	68.23
3	73	72.60	70.02
2	25	74.67	71.79
1	22	75.25	71.97

Table 1. Allocation of crowdworkers to quality categories and corresponding Control OA (binary) and Label OA (multi-class).

Table 1 shows, that the number of high quality workers for this task is quite low since only about a quarter of all workers is assigned to *Category* 1 and 2. However, for the initialization of the AL pipeline it is essential to provide correct samples. Therefore, we only rely on workers of at least *Category* 2. The controlling assignments of these workers achieve an OA of 74.67 %.

All points detected as labeled falsely by this second campaign are eliminated as well as all labels controlled by workers of *Category* 3 or worse. Of course, this procedure causes a loss of potentially correct labels given in the first campaign. However, for initialization only a small pool of correct samples for each class is required. By this control job the OA of the crowd labeling is increased from 56.44 % to 71.79 %. The corresponding confusion matrix in Figure 3 (b) demonstrates that the amount of confused labels between already mentioned classes is only decreased partially. Therefore, we decided to merge classes the crowd seems not to be able to distinguish reliable. Classes of middle and high vegetation, namely *Fence/Hedge*, *Shrub* and *Tree* are merged to *Mid- and High Vegetation*. Due to the sparse distribution of class *Powerline* in the V3D dataset we also omit class *Powerline* and unite it with class *Roof*. So, we are focusing on classes which have proven to be suitable for acquisition by crowd. By this, we increase the OA for the crowd labeling to 88.59 % and therefore provide a reasonable training dataset for initialization of the AL pipeline.

The last type of crowd job is labeling of most informative samples queried in the AL procedure (section 2.3.3). In each iteration step we use an experimentally determined batch size of 306 points having highest VE, which are offered to the crowd, so that a total of 34 jobs are available (9 points per job). For controlling the crowd two mechanisms are applied. First, we are again including 4 control samples (see section 2.3.3). In order to get a high quality labeling result we only consider submitted jobs from workers of *Category* 1. Therefore, we are rejecting and automatically reposting those of lesser quality. Second, every point is labeled three times by different crowdworkers. From this multiple acquisition the true label is afterwards inferred via majority vote, which has proven as a simple but robust measure (Zhang et al., 2016). So, only if two or more workers labeled a point identically, we add it to the training pool. Following this strategy, the resulting labels are not only given by a single person as often done by one expert. Although in crowdsourcing the problem of label bias is as well present for individual workers, it can be mitigated (Wauthier and Jordan, 2011).

All labeled points gained in the course of the AL process are afterwards compared to the reference, resulting in an OA of 88.87 %.

The corresponding confusion matrix in Figure 3 (c) outlines confusion between classes *Impervious Surface* and *Low Vegetation*. This is due to the fact that the exact border between these classes is difficult to spot in the point cloud and exact labeling is even demanding for an expert. A second problem an expert can overcome due to his knowledge and interpretation ability of the data, are shadowed surfaces. The crowd on the other hand tends to select *Low Vegetation* for shadowed street points. Moreover crowdworkers often mistake points lying exactly on the border between *Roof* and *Façade*. This mismatch is rather an issue of definition. Furthermore points belonging to *Mid- and High Vegetation* are often allocated to other classes. It seems that crowdworkers assign all points they can not clearly match to a specific class to *Mid- and High Vegetation*. This behavior is however understandable since in such cases the local distribution of the neighboring points is typically extremely sparse and additionally often characterized by an irregular point distribution typical for vegetation. Labeling such points is as well highly demanding for an expert who may only determine the correct class by referring to corresponding imagery.

At this point, we want to state that the afore-mentioned accuracy can be reached after manual inspection of the reference data of V3D and correction of in our opinion falsely labeled points especially occurring at class borders (labels of 308 points were corrected). According to the authors of Rottensteiner et al. (2014) V3D has been labeled by one student assistant. So as a side effect, our approach can also be used in order to verify labels of a given point cloud. For the sake of completeness, using the original GT an OA of $79.66\,\%$ is reached.

3.2 Statistics of all Campaigns

Besides the quality of the results, another critical factor is time required for a task and total running time of a complete campaign which is as well depending on payment as shown in Walter and Soergel (2018). The statistics of all campaigns are presented in Table 2.

Campaign	mean time/ task [min]	campaign time [h]	paym./ task [$]
Point Picking	6.42	23.60	0.10
Control Labeling	4.70	17.20	0.12
Iteration ∅	2.53	14.80	0.10 + 0.05

Table 2. Statistics of all conducted crowd campaigns.

According to the mean time a worker spent on a task, the picking job is as expected the most time-consuming since here crowdworkers have to navigate within the point cloud to search for points of each class. A striking fact is that workers spent more time on the controlling job than on labeling complex points. This indicates that labeling a point cloud from scratch may be easier or more intuitive than checking a given label. Since operating time per worker for all AL iteration steps shows a standard deviation of only 0.2 min, in terms of time spent on labeling complex points, points of all classes have the same level of complexity. The mean time of 14.8 h for such a campaign may be decreased by posting jobs not only to *Top Performers* but to a bigger group. Due to a quite harsh quality control mechanism we decided to provide a bonus of $0.05\,\$$ to all workers passing all tests (see section 2.3.3) as monetary incentive. In total we have spent $144.40\,\$$ $(100 \cdot 0.10\,\$ + 100 \cdot 0.12\,\$ + 102 \cdot 8 \cdot 0.15\,\$)$ for establishing the training dataset. Furthermore we evaluate the countries of origin of all crowdworkers who have participated in our campaigns.

Figure 4 validates the findings of Hirth et al. (2011), stating that the crowdworkers using *Microworkers* are mainly situated in low wage countries.

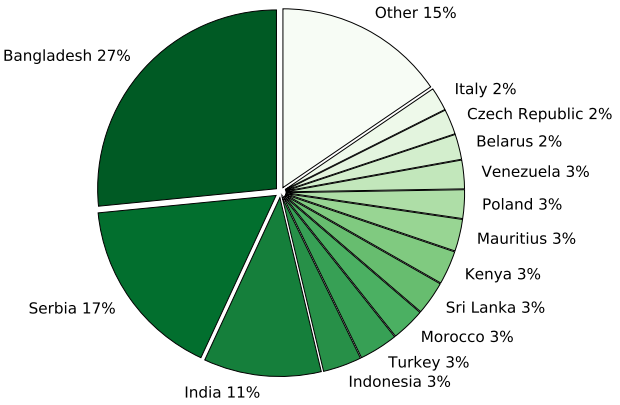


Figure 4. Countries of origin of all crowdworkers participating in our campaigns.

3.3 Results of AL Pipeline

Figure 5 presents the actual class affiliations of queried points within the AL process. As previously mentioned, most informative samples are located along class borders. This is also implicitly shown in Figure 5. For instance, in the first iteration step, points representing *Low Vegetation* and *Impervious Surface* are queried. These two classes are adjacent both in feature and object space. This means the classifier wants to learn how to distinguish between these two classes and where to put the decision boundary between them in feature space. This behavior of selecting points lying on borders between classes holds for all iteration steps. The classifier seems to especially focus on distinguishing between two classes per step, which are up to now difficult to separate, for example *Roof* vs. *Façade* in the second iteration step and *Car* vs. *Impervious Surface* in the third step. This is observable until the fourth iteration step. Afterwards the underlying classes of queried points are more equally distributed. This means, that the machine knows how to distinguish between different classes in general and now has to focus on especially complex samples which are spread among classes.

In basic AL, all points queried by the classifier are labeled by an omniscient oracle giving a perfect labeling. Since in our case this oracle is given by the crowd, as an inherent challenge of our approach, our labels at each step are not perfect. This can be due to poor crowdworkers, various interpretation of data by the crowd mainly based on the crowdworkers' cultural origin, suboptimal GT inference or simply complex-to-label points even an expert cannot label with high confidence. From section 3.1 we already know that the OA of the crowd is $88.87\,\%$. In Settles (2009) the question was risen to what degree AL works with noisy labels. In the context of this study, we try to provide an answer for AL in geospatial 3D point interpretation.

Figure 6 represents the progress of F1-scores (Goutte and Gaussier, 2005) of all classes for initialization and every iteration step evaluated on the test site of V3D, which is completely unknown to the classifier. It can be seen that the gradual addition of informative points labeled by the crowd to the training pool initially leads to a rapid increase in OA, which flattens out more and more until iteration step 8. This behavior together with the

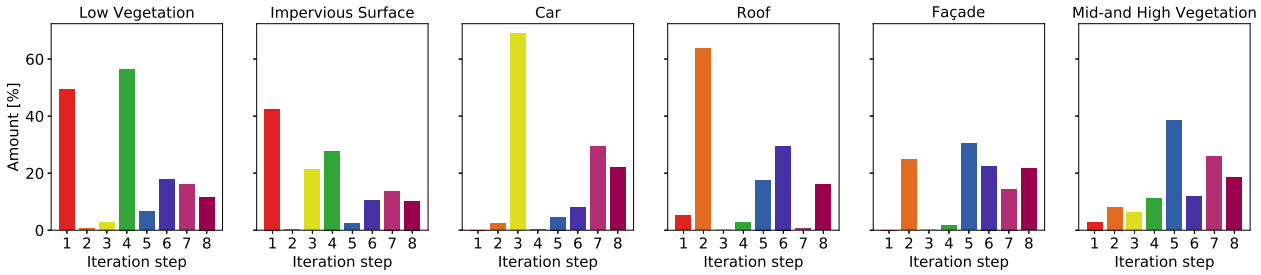


Figure 5. Relative amount of selected points per class and iteration step queried within the AL pipeline.

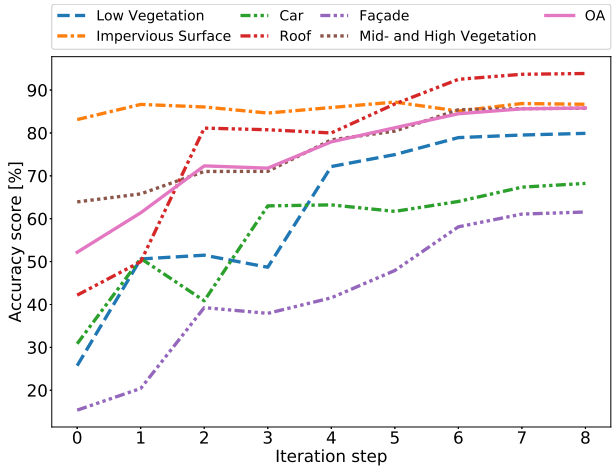


Figure 6. Achieved F1-scores per class (*dotted)* and OA (*solid*) of the RF for the test set of V3D in the process of 8 iteration steps.

fact that the classifier only selects remaining classification uncertainties spread among classes, as already observed in Figure 5, causes us to terminate the iteration. Altogether, the OA has been optimized starting from a quite low value of $52.19\,\%$ only using samples of the initialization to an OA of $85.82\,\%$ including most informative samples. It can be clearly seen that in every iteration step especially classes for which labeled points are added to the training pool profit, but appending labels always leads to an impact on all classes. A remarkable behavior is observable in iteration step 3 where OA decreases marginally. Here, mainly labels for class *Car* are queried so that the F1-score of this class increases significantly. Since class *Car* is underrepresented in the point cloud, the performance of other classes is reduced because now in the training dataset this class is overrepresented.

In order to demonstrate the efficiency of our approach, we oppose the result of the automated classification via RF using our acquired sparse training pool to the outcome of using the complete training dataset of V3D. Table 3 shows that relying on our highly informative instances instead of the complete V3D training set leads to a difference in OA of less than 3 percentage points after only 8 iteration steps. This roughly holds for the F1-scores of all classes equally. Since the RF was already involved in setting up the training pool, we demonstrate general applicability of our derived training dataset by using it to train a deep neural network. For this, we apply a 3D submanifold sparse convolutional network (SSCN) as used in Schmohl and Soergel (2019). Although such networks usually rely on huge training datasets, Table 3 outlines, that this classifier performs quite similar to the employed RF for the sparse training set. In terms of efficiency, these results support our approach of relying on less labels but carrying valuable information due to their nature on lying close to the decision border in contrast to using a vast label pool of redundant typical samples.

We want to stress that the accuracies based on our crowd-based training pool are achieved without using any labels from the V3D benchmark and by only labeling $0.4\,\%$ of available 3D points of the V3D training site by the crowd. This classification result has been achieved by only giving the unlabeled training dataset and a budget of about $145\,\$$ as input.

4. CONCLUSION AND OUTLOOK

In this paper we have shown that extracting a sparse training pool providing most informative training samples can be acquired in a fully automated process by means of the crowd in combination with ML techniques. This is accomplished by a hybrid approach both incorporating automatic selection of these samples and manual labeling of them by the crowd. By using this dataset for training we have demonstrated that we can significantly reduce labeling effort without major loss in classification performance after only 8 iteration steps. However, for labeling points we have observed that the crowd can not meet arbitrary requirements of the

Classifier	Training set	F1-score [%]						OA[%]
		Low Vegetation	Impervious Surface	Car	Roof	Façade	Mid-and High Vegetation	
RF	V3D	82.55	91.84	69.85	95.01	62.67	86.15	88.42
	CB-AL	79.91	86.69	68.23	93.86	61.57	85.77	85.82
SSCN	V3D	82.48	91.16	75.15	94.89	61.53	87.29	88.39
	CB-AL	80.00	88.14	75.20	91.08	57.34	84.72	85.43

Table 3. Comparison of classification results on the test site of V3D when using the given V3D training dataset vs. using labels derived by our crowd-based AL approach.

employer. Especially classes representing vegetation are quite difficult for the crowd to differentiate. In our future work, we will further analyze the impact of such noisy labels on the classification process. Most issues in this study rise due to different time of acquisition of the point cloud and the image data. But since modern LiDAR sensors are commonly equipped with imaging systems as well, this problem is likely to be avoided in recent datasets. While the point density of the V3D dataset is often realized in national mapping, in UAV-based acquisition campaigns a far higher point density can be achieved (Haala et al., 2020). We assume that labeling such dense datasets is less demanding for crowdworkers and opens up the possibility of acquiring more detailed structures such as façade and roof information.

REFERENCES

Argamon-Engelson, S. and Dagan, I., 1999. Committee-Based Sample Selection For Probabilistic Classifiers. *Journal of Artificial Intelligence Research* 11, pp. 335–360.

Branson, S., Wah, C., Schroff, F., Babenko, B., Welinder, P., Perona, P. and Belongie, S., 2010. Visual Recognition with Humans in the Loop. In: *Computer Vision – ECCV 2010*, Springer Berlin Heidelberg, pp. 438–451.

Breiman, L., 2001. Random Forests. *Machine Learning* 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 Psych. Sci.* 6(1), pp. 3–5.

Cabello, R., 2019. Three.js [WWW Document]. URL: https://threejs.org (accessed Jan 15, 2020).

Cramer, M., 2010. The DGPF-Test on Digital Airborne Camera Evaluation – Overview and Test Design. *Photogrammetrie - Fernerkundung - Geoinformation* 2010(2), pp. 73–82.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Li, F. F., 2009. ImageNet: A Large-Scale Hierarchical Image Database. In: *Proceedings of 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: *Proceedings CIKM 2007*, ACM, New York, NY, USA, pp. 127–136.

Feng, D., Wei, X., Rosenbaum, L., Maki, A. and Dietmayer, K., 2019. Deep Active Learning for Efficient Training of a LiDAR 3D Object Detector. *30th IEEE Intelligent Vehicles Symposium*.

Gadiraju, U., Kawase, R., Siehndel, P. and Fetahu, B., 2015. Breaking Bad: Understanding Behavior of Crowd Workers in Categorization Microtasks. In: *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, ACM, pp. 33–38.

Goutte, C. and Gaussier, E., 2005. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In: *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 345–359.

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. Accepted for publication in: *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*

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

Hetmank, L., 2013. Components and Functions of Crowdsourcing Systems – A Systematic Literature Review. In: *11th International Conference on Wirtschaftsinformatik*, pp. 55–69.

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

Hu, X. and Yuan, Y., 2016. Deep-Learning-Based Classification for DTM Extraction from ALS Point Cloud. *Remote Sensing* 8(9), pp. 730.

Kellenberger, B., Marcos, D., Lobry, S. and Tuia, D., 2019. Half a Percent of Labels is Enough: Efficient Animal Detection in UAV Imagery Using Deep CNNs and Active Learning. *IEEE Transactions on Geoscience and Remote Sensing* 57(12), pp. 9524–9533.

Kölle, M., Laupheimer, D. and Haala, N., 2019. Klassifikation hochaufgelöster LiDAR- und MVS-Punktwolken zu Monitoringzwecken. In: *Publikationen der DGPF*, Vol. 28, DGPF, pp. 692–701.

Kovashka, A., Russakovsky, O., Fei-Fei, L. and Grauman, K., 2016. Crowdsourcing in Computer Vision. *Foundations and Trends in Computer Graphics and Vision* 10(3), pp. 177–243.

Mackowiak, R., Lenz, P., Ghori, O., Diego, F., Lange, O. and Rother, C., 2018. CEREALS - Cost-Effective REgion-based Active Learning for Semantic Segmentation. *BMVC 2018*.

Niemeyer, J., Rottensteiner, F. and Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. *ISPRS Journ. Photogramm. Remote Sens.* 87, pp. 152–165.

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: *Proceedings of NIPS 2017*, Curran Associates Inc., USA, pp. 5105–5114.

Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J. D., Breitkopf, U. and Jung, J., 2014. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. *ISPRS Journ. Photogramm. Remote Sens.* 93, pp. 256–271.

Schmohl, S. and Soergel, U., 2019. Submanifold Sparse Convolutional Networks For Semantic Segmentation of Large-Scale ALS Point Clouds. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* IV-2/W5, pp. 77–84.

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

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

Wauthier, F. L. and Jordan, M. I., 2011. Bayesian Bias Mitigation for Crowdsourcing. In: *Advances in Neural Information Processing Systems 24*, Curran Associates, Inc., pp. 1800–1808.

Weblabcenter Inc., 2019. Microworkers [WWW Document]. URL: https://www.microworkers.com (accessed Jan 15, 2020).

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

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

Yang, B., Luo, W. and Urtasun, R., 2018. PIXOR: Real-time 3D Object Detection from Point Clouds. In: *Proceedings of CVPR 2018*, pp. 7652–7660.

Zhang, J., Wu, X. and Sheng, V. S., 2016. Learning from crowdsourced labeled data: a survey. *Artificial Intelligence Review* 46(4), pp. 543–576.