2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Semantic Mapping of Construction Site from Multiple Daily Airborne LiDAR Data

Thomas Westfechtel[1], Kazunori Ohno[2,3], Tetsu Akegawa[4], Kento Yamada[4], Ranulfo Plutarco Bezerra Neto[4], Shotaro Kojima[4], Taro Suzuki[5], Tomohiro Komatsu[6], Yukinori Shibata[7], Kimitaka Asano[8], Keji Nagatani[1], Naoto Miyamoto[2], Takahiro Suzuki[2], Tatsuya Harada[1], Satoshi Tadokoro[4]

***Abstract*— Semantic maps are an important tool to provide robots with high-level knowledge about the environment, enabling them to better react to and interact with their surroundings. However, as a single measurement of the environment is solely a snapshot of a specific time, it does not necessarily reflect the underlying semantics. In this work, we propose a method to create a semantic map of a construction site by fusing multiple daily data. The construction site is measured by an unmanned aerial vehicle (UAV) equipped with a LiDAR. We extract clusters above ground level from the measurements and classify them using either a random forest or a deep learning based classifier. Furthermore, we combine the classification results of several measurements to generalize the classification of the single measurements and create a general semantic map of the working site. We measured two construction fields for our evaluation. The classification models can achieve an average intersection over union (IoU) score of 69.2% during classification on the Sanbongi field, which is used for training, validation and testing and an IoU score of 49.16% on a hold-out testing field. In a final step, we show how the semantic map can be employed to suggest a parking spot for a dump truck, and in addition, show that the semantic map can be utilized to improve path planning inside the construction site.**

I. INTRODUCTION

With the current swift advancement of autonomous vehicles, more and more fields are investigating the employment of autonomous technologies. One major requirement for autonomous vehicles to operate safely in complex environments are detailed semantic maps. While in public areas the semantics mostly follow strict rules defined by the guidelines and laws, construction sites have their own rules of layout. The road, construction vehicle parking area, personal vehicle parking area, sediment area, and construction material area are not strictly defined but are decided by a multitude of factors (i.e. the workers, the working area shape, construction vehicle motion).

Understanding the semantics of an environment is essential for human workers and autonomous construction vehicles. A good grasp of the layout and the rules of the construction site are necessary for a smooth and safe operation. Furthermore, by following the rules, autonomous vehicles can behave in a way that is easily understandable and predictable by human workers. This is essential for the acceptance of automated vehicles in a shared human-robot workspace and for a successful human-robot collaboration. However, in order to understand the semantics of construction sites, it is necessary to observe the site over a longer period of time. As the working site is in active use, the snapshot of a specific time does not necessarily reflect the underlying semantics. For example, in Fig. 1 many backhoes can be found in the middle of the construction field, blocking off the left side of the construction field. As this was specific for a single measurement, after fusing multiple daily data, these backhoes do correspond to the underlying semantic region for backhoes.

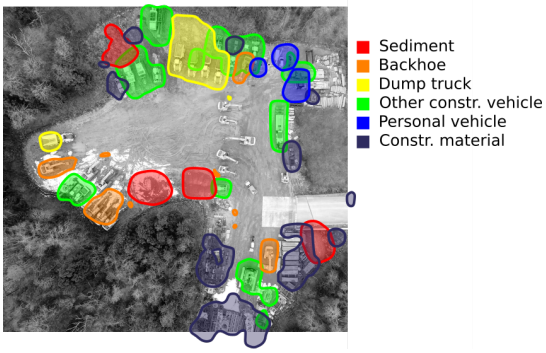


Fig. 1: Extracted semantic regions for a construction site overlayed to a picture of the construction site.

In this paper, we propose a method to autonomously extract semantic regions from a construction site. A drone equipped with a Velodyne VLP-16 LiDAR and six Global Navigation Satellite System (GNSS) antennas is employed to generate a digital elevation map of the construction site [1]. From these measurements, we estimate the ground plane and extract connected regions above the ground. As these regions do not necessarily represent a single object, we introduce a competing subclustering method. The subclustering method further disentangles the connected regions and chooses the combination of subclusters that fit best to the trained classifier model (we use the confidence score of the classifier as fitting score). For the classification we compared three

[1] T. Westfechtel, K. Nagatani, T. Harada are with The University of Tokyo thomas@mi.t.u-tokyo.ac.jp
[2], K. Ohno, N. Miyamoto, Takahiro Suzuki are with New Industry Creation Hatchery Center, Tohoku University, Sendai, Japan
[3] K. Ohno is with the RIKEN Center for Advanced Intelligence Project
[4] T. Akegawa, K. Yamada, RP. Bezerra Neto, S. Kojima S. Tadokoro are with the Graduate School of Information Sciences, Tohoku University, Sendai, Japan
[5] Taro Suzuki is with Chiba Institute of Technologies
[6] T. Komatsu is with Kowa Tech. Co. Ltd.
[7] Y. Shibata is with Satoh Construction Co. Ltd.
[8] K. Asano is with Sanyo Tech. Co. Ltd.

classifier: two deep-learning based classifier and a random forest classifier. In the case of random forest, we employ the decision distribution from the single trees as confidence score.

In a next step, we fuse the classified objects over multiple days to create a semantic map, which shows the different areas of the construction site. In this way, we can identify areas that are commonly used for construction material, parking place for construction vehicles, dump trucks and personal vehicles as shown in Fig. 1. Furthermore, we show that the semantic map can be employed for suggesting parking areas for different classes of vehicles and can provide a more general map for path planning.

II. RELATED WORKS

Semantic mapping is an important research topic in data engineering, artificial intelligence and robotics. Semantic maps are useful in many cases. They can provide a dense representation of the environment that allows humans quickly grasp relevant information. In the case of robotics, semantic maps enable robots to have a better understanding about its environment. Specific areas in the map are given additional high-level knowledge that can be exploited by the robot to act and interact more reliably with its environment, which in turn enables the robot to act in a more human-understandable way.

Yang et al. [2] presented a method for building 3D semantic maps from stereo image streams. The researchers used CNN-based image segmentation in combination with high order conditional random fields to build geometric maps that enable mobile robots to not only avoid obstacles, but also to recognize objects for high-level tasks. Kochanov et al. [3] followed a similar approach, but placed emphasis on temporal tracking and updating dynamic (moving) objects in a temporally and spatially consistent semantic 3D map of the surroundings. As autonomous vehicles need to operate in a dynamic environment, they need to have a persistent model of the static surrounding as well as keep track of dynamic objects. Westfechtel et al. [4] presented a method to autonomously enrich street level maps with parking spot information. In their work, parked vehicle were detected through a bird's eye view elevation map generated from LiDAR data. In a graph-based approach the detections were used to infer the underlying parking lot structure, allowing the robot to map parking spots in the environment. Another research in this field was carried out by Sünderhauf et al. [5]. The researchers combined semantic meaning in the form of object classifications with the point- or mesh-based representations of the objects and the environment in order to build semantic enriched environmental maps. Such maps provide robots with knowledge about their environment in form of semantic meaning and at the same time provides an accurate geometrical representation that enables the robot to interact with the objects.

Recently more and more researchers investigate in using neural networks for the semantic segmentation of point cloud data. Some of these methods are based on voxel-based 3D convolutions ([6],[7]), view-based 2D convolutions, ([8],[9], or work directly on the point cloud data ([10],[11]). A detailed overview can be found in [12].

Furthermore, as a single measurement is a snapshot of the current usage of the construction field at a single time, we fuse the semantics of several days to obtain more general semantics. Therefore a large part of this work is object segmentation and classification.

Object segmentation and classification has been subject of studies for many different cases and applications. In this review, we focus on the case of employing depth data.

Serna and Marcotegui [13] presented a method to classify urban objects from point clouds. In their work, the ground plane was estimated and objects were detected as discontinuities of the ground. A support vector machine (SVM) with geometrical and contextual features was used for classification of the objects. Similarly Roynard et al. [14] classified objects from depth data gathered by a vehicle equipped with a rear mount LiDAR. The researchers estimated and removed the ground plane and then clustered connected objects. Different descriptors for a random forest were compared to predict the semantic labels. Bogoslavskyi and Stachniss [15] employed angles calculated from a range image acquired by a 3D-LiDAR mounted on top of a vehicle to detect the ground plane. In a subsequent step, objects were segmented in the range image using an angle-based approach. The research focused on a real-time segmentation and did not classify the detected objects.

From these related works, the scope and difficulty of semantic mapping and scene understanding becomes apparent. In our case, we want to add semantic information to a map of a construction site, i.e. areas for different construction vehicles, areas for parking personal vehicles and areas for construction material. The usage of drones gives a wide overview of the whole construction field, but at the same time a limited viewing angle and a limited resolution. In addition, we have very limited data. As these sites are in active use it is difficult to gather data from multiple construction sites over several days. This limited data also prevents us from directly employing a deep learning algorithm (as most of the previous research did).

Instead, we chose to extract objects at instance level first and train a classifier for these instance level. The extraction at instance level separates the object from the background and therefore grants more generability for the classifier. For the object segmentation, we follow the common approach of estimating the ground plane [16] in a first step and cluster the points above the ground plane to get segmented regions at an object level. In order to improve the ground plane estimation, we introduce a consistency ground check. Furthermore, we introduce a competing subcluster algorithm to separate clusters containing more than one object.

For the classification, we investigated two methods. In one approach, we create a feature vector for each object and train a random forest similar to [14]. In the second approach, inspired by [8] and [9], we create bird's eye view images for the classification. However in contrast to the aforementioned

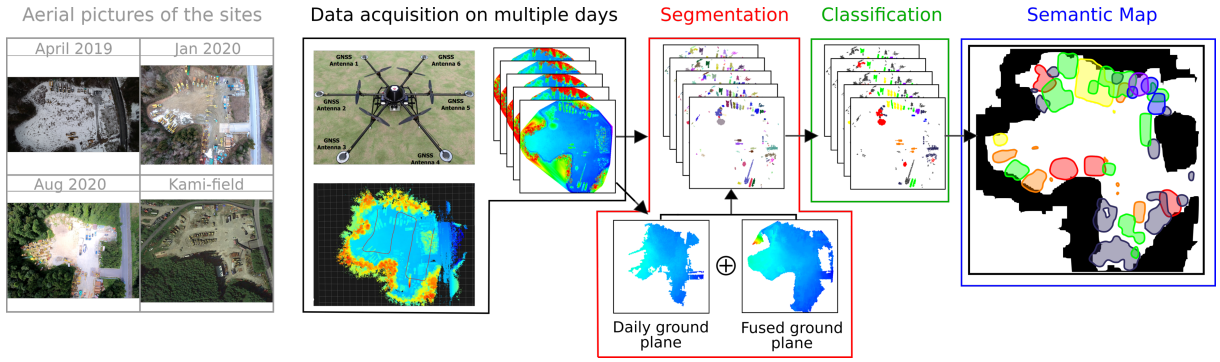


Fig. 2: Flowchart of our proposed algorithm. An UAV with multiple GNSS receivers is used to measure multiple daily DEMs of the construction site. We estimate the ground plane. Using the ground plane we extract object from the daily data. Each object proposal is classified into one of seven classes. Potential fields are created for the detections of each class and day. By fusing the potential fields over multiple days a semantic map of the construction site is generated.

methods, we create an image for each object result of our clustering algorithm and only classify the image without the task of object detection. The separation of the object detection and classification is done deliberately due to the small amount of data.

The main contributions of this work are the following:

- A method that extracts semantic regions from multiple daily airborne data for construction sites. We show that the semantic map can be used for suggesting a parking spot and to support a path planning algorithm to avoid frequently occupied areas.
- An algorithm for generating competing subclusters to disentangle clusters containing more than one object. The competing subclusters are scored by how well they fit to the trained model.
- A deep learning approach to classify objects in the point cloud data by transforming them into three channel images (equivalent to RGB) as well as a random forest approach.

III. METHODOLOGY

For our semantic mapping algorithm, different core technologies were combined. A coarse flowchart of our processing stream can be seen in Fig. 2. The algorithm can be divided into three parts.

For the generation of the digital elevation map (DEM) an UAV equipped with a LiDAR and six GNSS antennas was employed. From the GSNN data, precise position and pose information of the LiDAR is obtained and used to generate a precise DEM from the LiDAR's data [1]. The resulting DEM has height values for each cell in a grid size of $25\,\mathrm{cm}$, where each second row consists of one height value and each other row consists of three height values (maximum, minimum and mean height). For processing purposes we transform the DEM into a point cloud where each height value is represented by one point.

In the first part, the point cloud is segmented at object level. For the segmentation, we extract the ground plane for each measurement. Ground measurements that vary too much between the daily measurements are removed from the ground plane and the ground height is estimated for cells without ground measurements by fitting a plane to the surrounding points of the ground plane. Connected regions above the ground plane are extracted, as these connected regions may represent more than one object, we generate competing subclusters for each connected region.

In the second part, each object proposal is classified into one of seven object classes. For the classification we employ two methods. The first estimates a feature vector for each segmented object and employs a random forest for classification. The second method transforms each object into a three channel image (equivalent to RGB) and we train two convolutional neural networks (CNN) for classification. The confidence score of the classification is also employed to score the competing subclusters and choose the subclusters that best fit to the trained model.

In the last part, the segmentation results are fused in a probabilistic way over multiple daily measurements to generate a semantic map of the environment.

A. Ground plane

The main idea for segmenting the objects is to extract connected objects above the ground plane. In order to get the ground plane, we extract the largest connected plane in the measurement using a region growing algorithm where neighboring points have to have an inclination below $30°$. We chose this value as it is slightly below the angle of response for dry sand ($34°$), and we want to detect the sediment piles as object. However, small sediment piles may have a smaller inclination angle and be detected as ground. Therefore, we further test if the ground height is consistent between the measurements. In particular, we remove all ground plane cells that differ more than three times the standard height deviation. This leads to a ground plane estimation with many "holes" due to objects blocking the measurement of the ground plane. By holes, we refer to areas with no height information of the ground that are completely surrounded by the ground plane. We estimate height measurements for these holes, by fitting a plane through the surrounding height measurements of the ground plane. In a last step, we fuse

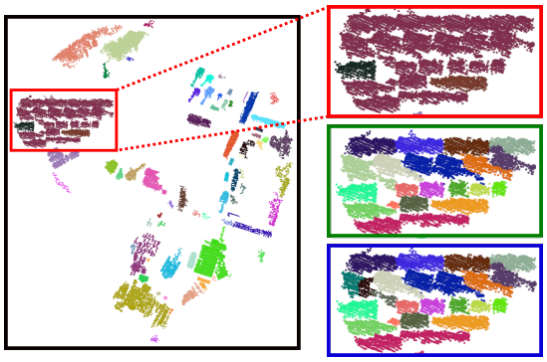


Fig. 3: Results for the object extraction process (left side). The right side depicts the subclustering results for the framed area. The result before subclustering is framed by red (top). The manually labeled best subclusters are framed by green (middle) and the predicted subclusters are marked by blue (bottom).

the measurements by taking the average for each cell. For the object segmentation, the ground plane for that specific day is preferred for calculating the height above ground as it represents the most reliable data for that day (as the ground is mostly muddy, the height may vary based on weather and usage). We employ the fused ground plane only if the daily ground plane does not have a height measurement for a cell but the fused one does.

B. Object segmentation

Using the ground plane of the specific day and the fused ground map, we extract the points belonging to objects. Points $50\,\mathrm{cm}$ above the ground plane are extracted from each daily data. We chose the threshold of $50\,\mathrm{cm}$ as it proved to be a good trade-off between robustness against ground changes and measurement noise while extracting most points that belong to objects. The extracted points are clustered through a simple connectivity check in the grid cell, meaning extracted points in neighboring grid cells are clustered together. This clustering method achieved good results for most object instances. However as can be seen in Fig. 3 it fails in some cases where objects have little physical distance. To overcome this problem, we introduce a height-based competing subclustering method.

C. Competing subclustering method

As can be seen in Fig. 3, the euclidean clustering does fail to produce adequate object-level segmentation when objects are close-by. However it is hard to find a good deterministic rule to judge whether a cluster contains a single or multiple objects. Our main idea is that if we can find subclusters that fit better than the original cluster to our trained classifier, it is likely that the original cluster contains more than one object, i.e. the subclusters. We use the confidence score of our classifier as fitting score.

However, in order to make this idea work, we need to find subclusters at object-level. To obtain these subclusters at object-level we introduce a height-base iterative subclustering. In this process, the point with the least height above

Algorithm 1 Object Segmentation Algorithm

```
EC := Euclidean Clustering
function SUBDIVIDECLUSTER(Cluster C, Tree T)
    Sort points P_C ∈ C by height above ground
    foundSubClusters = false
    while |C| > 40 AND not(foundSubClusters) do
        Delete point with lowest height from C
        Employ EC on C to extract k subclusters C'_{j=1:k}
        (with condition of |C'_j| > 20 ∀j)
        if k > 2 then
            foundSubCluster = true
            for j = 1 : k do
                Add C'_j as node N_j to tree T
                subdivideCluster(C'_j, N_j)
procedure MAIN
    Extract all points P_obj > 50 cm above ground
    Extract clusters C_i from P_obj via EC
    Create a tree T_i with C_i as stem ∀i
    subdivideCluster(C_i, T_i) ∀i
```

the ground is iteratively removed from the starting cluster and an euclidean clustering is performed on the remaining points. If the clustering process finds more than one cluster (where each cluster has to have at least 20 points), we interpret the result as one possibility to subcluster the starting cluster. We add the points that were removed in the process to their nearest subcluster. This process of subclustering is then repeated for the discovered subclusters. The algorithm can also be seen in Alg. 1. This recursive process can be visualized in a tree structure with the original cluster as trunk (see Fig. 4a).

The tree structure visualization is further useful for scoring the competing subclustering candidates. Each point of the cluster can only belong to one object, which means that a tree node and its children are exclusive to each other. Therefore we can simply compare the confidence score s_{conf} of each node to the confidence scores of its children.

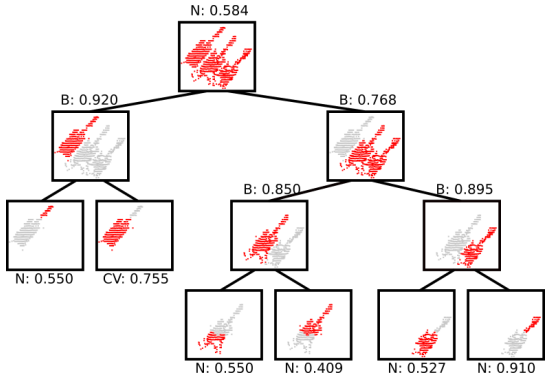
Starting from the leafs, all children $t_{i,j}$ of the same parent node t_i are compared to it, and the parent node is updated with the winning configuration (if the children configuration wins).

$$\tilde{s}_{\mathrm{conf}}(t_i) = \max(s_{\mathrm{conf}}(t_i), \frac{1}{j}\sum_{j} s_{\mathrm{conf}}(t_{i,j})) \qquad (1)$$

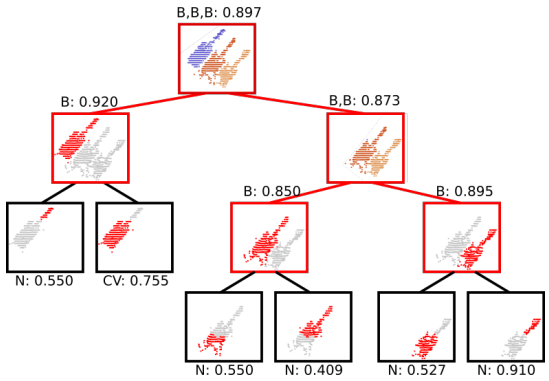
This process is run iteratively until the trunk of the tree. The process can be seen in Fig. 3. In the top subfigure the constructed tree can be seen, the bottom subfigure displays the result after calculating the best fitting score. We further introduce a weight vector for the classes. The weight vector is estimated by maximizing the accuracy (minimizing $1-$accuracy) on the validation set using the Levenberg-Marquardt algorithm.

D. Object classification

After the single objects are segmented, we classify them into one of seven classes. For the classification, we compare



(a) Subclusters (red points) at different levels with their predicted class and confidence score.



(b) Results of the decision process. The three subclusters classified as backhoes have the highest combined confidence score.

Fig. 4: Subclustering process and decision process. N:= Noise, B:= Backhoe, CV:= Construction vehicle.

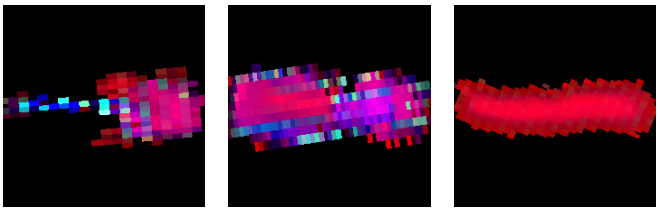
TABLE I: Composition of feature vectors.

Type	Name	Dim
F_{glob}	Eigenvalues	3
	Height, width, length	3
F_{height}	Height bin feature vector	19
F_{angle}	Angle bin feature vector	19
F_{dim}	Height (max, mean, var) along object's length	15
	Width (max, mean, var) along object's length	15

two methods. For the first method, we generate a feature vector for each object and train a random forest classifier. We chose a random forest classifier as it yields good results for sparse training data. For the second method, we transform the points of each object into a 3 channel image (equivalent to RGB) and finetune two pretrained convolutional neural network (CNN) classifier.

1) Random Forest classifier: For the random forest classifier, we create a feature vector consisting of 4 categories. In order to remove the dependency of the alignment of each object, we align each object along the eigenvector of their largest eigenvalue of the object's 2D projection. We chose the 2D projection as we wanted the height axis to remain the same. The composition of the features can be seen in Tab. I.

F_{glob} describes the object as a whole. It consists of the eigenvalues estimated by principal component analysis as well as the maximum height, width and length of the object.



(a) Created image for a backhoe. (b) Created image for a dump truck. (c) Created image for a sediment pile.

Fig. 5: Examples for object images created through the transformation. 1st channel is mapped to the R, 2nd to G, and 3rd to B channel

F_{height} is a height bin feature vector. At first, the height difference to the ground plane for each point is calculated. According to the height difference, each point is placed into a bin, where each bin spans a height range of $25\,\mathrm{cm}$, with the lowest bin starting at $50\,\mathrm{cm}$ (the threshold for extracting points during the object segmentation). The feature vector is then normalized to a sum of 1.

F_{angle} describes the distribution of angles of an object. For each point of the object the yaw and pitch angle is estimated. We require each point normal to have a positive pitch angle, meaning the z component of the point normal is ≥ 0. Furthermore, we project the yaw angle into a range of $0°$ to $180°$ to account for the ambiguity of aligning the objects along the eigenvector of their largest eigenvalue. We subdivide the range of the yaw and pitch angle into five divisions of equal range ($36°$ for yaw and $18°$ for pitch). Each estimated point normal is put into a bin with respect to the respective division of the yaw and pitch angle. The combined bin feature vector consisting thus of 25 bins. Following the construction of the height bin feature vector, we normalize the angle bin feature vector to a sum of 1.

F_{dim} is a feature vector describing the height and width of an object along its lengths. For this feature vector we divide the object into five equidistant parts along its length. For each part we calculate the maximum, mean and covariance of their height, as well as of their width. This results in 15 features for the height and 15 features for the width respectively.

2) CNN classifier: Beside the random forest classifier, we also compare two convolutional neural network (VGG19 [18] and Xception [19]). Due to the grid structure of our data, we choose an image-based approach. The major advantage of transforming the point cloud data into an image is that this enables us to employ networks that have proven to be effective in the image domain as well as using the weights from the network trained on ImageNet as our initial weights.

For the transformation of the point cloud of each object into an image, we first align each object along the eigenvector of their largest eigenvalue of the object's 2D projection (as was done for the random forest). The three channels are then filled according to the following:

- 1st-Channel: Maximal height above ground plane
- 2nd-Channel: Maximal height difference within the cell
- 3rd-Channel: Pitch angle of the point normal of the

(a) Detected construction vehicles on a single day.

(b) Potential field for construction vehicles on a single day.

(c) Potential field for construction vehicle over multiple days.

(d) Predicted area for construction vehicles.

Fig. 6: Semantic area extraction for the example of construction vehicles.

highest point

Each pixel of the image is filled with their closest point of the object's projection (or background). This is done to adequately populate the image, otherwise only few pixels would be colored (which would lead to the network not converting during training).

It has to be noted that there are three height values only for each second row in the grid. Other measurement only consist of one height value, therefore the 2nd-Channel is only partially filled. Some examples of resulting pictures from this transformation are displayed in Fig. 5.

E. Semantic map generation

Combining the detection of multiple daily data, we create a semantic map that displays the areas for each class. For example, areas where private vehicles are frequently parked would make good recommendations for finding a parking spot. In order to extract these areas, for one measurement i we create for each object o of a specific class c a potential field F around it that decreases with distance d (see Fig. 6b) (for the class of construction vehicles, see Fig. 6a).

$$F_{c,i,o}(d) = \begin{cases} 1 & d \leq 2.5\,\mathrm{m} \\ \frac{5\,\mathrm{m}-d}{2.5\,\mathrm{m}} & 2.5\,\mathrm{m} < d \leq 5\,\mathrm{m} \\ 0 & 5\,\mathrm{m} < d \end{cases} \tag{2}$$

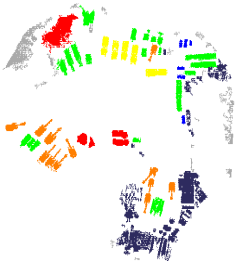
The potential fields of all objects of a class are added up for each measurement and capped at 1. The function G calculates the distance d between an object o and x, y coordinates. The set of all objects of a specific class is marked by O_c.

$$F_{c,i}(x,y) = \min(\sum\nolimits_{o \in O_c} F_{c,i,o}(G(o,(x,y)),1) \tag{3}$$

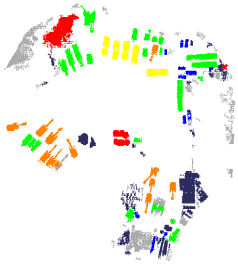
The potential fields of multiple days are averaged over all measurements I (see Fig. 6c).

$$F_c(x,y) = \frac{1}{||I||} \sum\nolimits_{i \in I} F_{c,i}(x,y) \tag{4}$$

To extract the semantic regions we employ a threshold on the accumulated potential fields (as shown in Fig. 6d). With more available data, we plan to adjust the weight to prioritize newer measurements.



(a) Manually labeled ground truth data of the construction site.



(b) Predicted results.

Fig. 7: Example result of the competing subclusters and classification, colors are according to the legend in Fig. 1, the class 'noise' is shown in grey.

TABLE II: Comparison of IoU-score between Random Forest and VGG19 and the combined method. The results are for measurement 4. Measurements 1,2,5,6 were used for training and measurement 3 was used for validation.

Object Class	Random Forest	VGG19	Xception
Sediments	0.5714	0.6000	**0.8333**
Backhoe	0.9286	**1.000**	**1.000**
Dump truck	0.800	**1.000**	**1.000**
Construction vehicle	0.7917	0.8182	**0.8500**
Personal vehicle	**0.5000**	0.2500	**0.500**
Construction material	0.6250	0.6500	**0.7692**
Noise/ Other	0.6429	0.6207	**0.7692**
Average	0.6942	0.7056	**0.8174**

IV. EXPERIMENTS AND RESULTS

For our experiments, we acquired data on six days over a time span of 10 months from a construction site near Sanbongi, Japan, as well as two measurements from a second construction field (Kami-field). A UAV equipped with a LiDAR sensor was used for data acquisition. We employed our object segmentation and subclustering for each measurement and manually labeled the objects and best subclusters to create the ground truth data. Four days of the Sanbongi data were used for training the random forest and the two CNNs, one day's data was used for validation and one day's data for testing. We cross-validated our model by rotating the days chosen for training, validation and testing. The measurements from the Kami-field are used solely for testing to assess the generability of our method. For the random forest, we employ the random forest classifier of

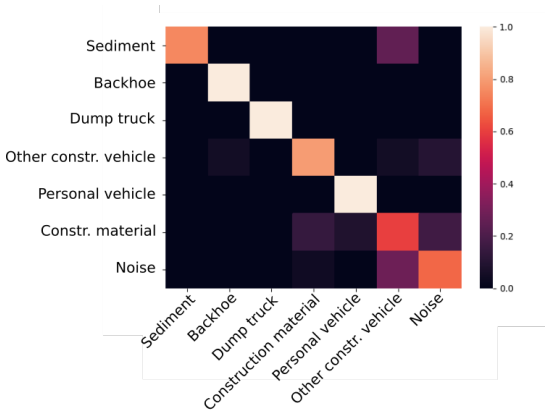


Fig. 8: Confusion matrix using the Xception classifier for measurement 4.

TABLE III: Comparison of IoU-score between Random Forest, VGG19 and Xception. The other five days beside the testing data are split into four days for training and one day for validation (for Sanbongi). The second construction field (Kami-field) was only used for testing, the data was not used for training.

Test Data	Random Forest	VGG19	Xception
Sanbongi 1	0.4603	0.4189	**0.5251**
Sanbongi 2	0.7527	0.8180	**0.8320**
Sanbongi 3	0.7129	**0.7833**	0.6072
Sanbongi 4	0.6942	0.7056	**0.8174**
Sanbongi 5	**0.7059**	0.6621	0.6811
Sanbongi 6	0.6185	**0.6902**	0.6894
Average	0.6574	0.6797	**0.6920**
Kami 1	**0.5025**	0.4931	0.3361
Kami 2	**0.4806**	0.4582	0.4543
Average	**0.4916**	0.4757	0.3952

the scikit-learn library with a size of 100 trees. For the CNN we chose the VGG19 [18] and the Xception network [19]. We randomly flipped the images horizontally and vertically during training and randomly rotated the images up to 7.5°. The network was initialized with the trained weights from Imagenet and trained until there was no improvement on the validation set for 20 continuous epochs.

Fig. 7 show the results of our algorithm of one measurement. On the left side (Fig. 7a) the annotated ground truth data and on the right side (Fig. 7b) the prediction (subclustering and classification) can be seen.

In order to evaluate the classification performance, we use the GT-split of the measurements. Tab. II displays the IoU scores for each object class for the random forest classifier and the two CNNs. It can be seen that the IoU-score is rather high for the classes backhoe and dump truck, while the IoU-score for construction material and noise is rather low. We believe that the discrepancy in result can be explained by the larger intra-class variance for the later classes. Fig. 8 displays the confusion matrix for the measurement.

Tab. III shows the IoU scores for each individual measurement. The CNN approaches outperform the Random Forest on the Sanbongi measurements. All of the classifiers have a drop in performance for the Kami-field. The drop for the deep learning approaches is considerably bigger than for the Random Forest, indicating that the CNN overfits the data of the Sanbongi field.

TABLE IV: Accuracy of correctly selected subclusters for the Sanbongi- (SB) and Kami-field (K) measurements. Left value in a box uses the estimated class weights, the right value weights all classes equal. We compare our method to the baselines of not using the subclustering process (No SC) and maximal subclustering (Max SC), i.e. the leafs of each tree.

Data	RF		VGG19		Xception		No SC	Max SC
SB 1	0.36	0.32	0.55	0.55	**0.64**	0.50	**0.64**	0.32
SB 2	0.64	0.64	0.67	0.58	0.55	**0.73**	0.64	0.18
SB 3	**0.71**	0.66	0.66	0.49	0.63	0.59	0.46	0.41
SB 4	0.78	0.80	0.69	0.65	**0.84**	0.63	0.55	0.22
SB 5	**0.61**	0.59	0.53	0.51	0.53	0.47	0.51	0.25
SB 6	**0.63**	0.56	0.44	0.30	0.63	0.54	0.46	0.24
Avg	0.62	0.59	0.59	0.51	**0.64**	0.58	0.54	0.27
K 1	0.46	0.41	**0.60**	0.39	0.38	0.33	0.39	0.33
K 2	0.57	0.56	**0.63**	0.45	0.52	0.49	0.42	0.39
Avg	0.52	0.49	**0.61**	0.42	0.45	0.41	0.41	0.36

Tab. IV shows the accuracy of the subclustering process. We added two baselines to our evaluation, not using the subclustering process (No SC) and maximal subclustering (Max SC), i.e. using the leafs of each tree. It can be seen that the class weights increase the performance in most cases. On average it increases the accuracy by 3%pts for RF, 8%pts for VGG19 and 6%pts for Xception. Using the weights all three methods outperform the baselines in most cases. It is also interesting to see that the no splitting baseline performs much worse on the Kami-field data, indicating that the Kami-field data is much more convoluted compared to the Sanbongi data.

Furthermore, the Sanbongi field is currently undergoing an extension. For this, the north-west part of the construction field has been vacated. We gathered three more daily data and generated a semantic map for the new measurements. The change of the semantics in the construction field can be seen in Fig 9.

We further fused the object classifications of all six days to generate a semantic map (see Fig. 10). From the results, we set a point within the common dump truck parking area as goal point for a dump truck entering the construction site. A path was planned from the entrance of the construction site to the suggested goal using a path planning algorithm [20]. If the algorithm only considers the obstacle grid map of a single measurement, the resulting path would cross frequently occupied regions. By also considering the semantic map the algorithm can avoid these areas, making the resulting path more general viable.

V. CONCLUSION AND DISCUSSION

In this work, we showed a method to create a semantic map from multiple daily data of a construction field. Connected objects above the ground plane are extracted from the daily measurements. We introduce a competing subclustering process to further disentangle connected objects that might represent multiple object instances. We compared a random

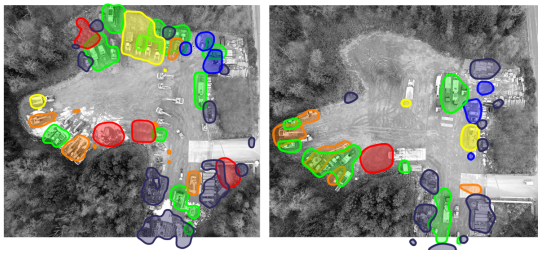


Fig. 9: Change of the semantics of the construction field. On the left side before and on the right side after the north-west part has been vacated for extending the construction field. Colors are according to Fig. 1.

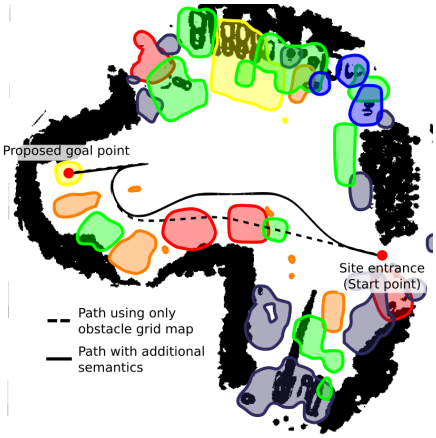


Fig. 10: Extracted semantic areas overlayed to an obstacle grid map of a single measurement. The semantics are used to suggest a parking spot for a dump truck and a path to the goal is planned. A path planned with only the obstacle grid map of a single measurement (dashed line) would result in crossing frequently occupied areas, these areas can be avoided when considering the semantic areas during the path planning (continuous line). Colors are according to Fig. 1.

forest with two deep learning classifier to classify the objects and simultaneously score the competing subclusters. Using the classification results from multiple daily data, we create a semantic map of the construction site. We further show that our algorithm is also applicable to other construction sites, though with a decrease in accuracy.

We showed how the semantic map can be employed to suggest a parking spot. In addition to the parking suggestion, the semantic map can support a path planning algorithm to avoid frequently occupied areas, resulting in a more general viable path, that is valid not just for a single day measurement.

In future work, we want to combine the semantic map with the prediction of backhoe behavior to automate a dump truck inside the construction site for loading sediment. Furthermore, we plan to measure the construction field on a more regular basis and gain more information based on the daily changes (i.e. height change of the sediment pile, frequency of occupancy, ...).

ACKNOWLEDGMENT

This research has been partially supported by CREST No. 14532298 and NEDO No. 18065741.

REFERENCES

[1] T. Suzuki, D. Inoue, Y. Amano ”Robust UAV Position and Attitude Estimation using Multiple GNSS Receivers for Laser-Based 3D Mapping” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019

[2] S. Yang, Y. Huang, S. Scherer ”Semantic 3D Occupancy Mapping through Efficient High Order CRFs” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017

[3] D. Kochanov, A. Ošep, J. Stückler, B. Leibe ”Scene Flow Propagation for Semantic Mapping and Object Discovery in Dynamic Street Scenes” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016

[4] T. Westfechtel, K. Ohno, N. Mizuno, R. Hamada, S. Kojima, S. Tadokoro ”Parking Spot Estimation and Mapping Method for Mobile Robots” in IEEE Robotics and Automation Letters (RA-L), 2018 Jun 22;3(4):3371-8.

[5] N. Sünderhauf, T.T. Pham, Y. Latif, M. Mailford, I. Reid ”Meaningful Maps With Object-Oriented Semantic Mapping” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017

[6] D. Maturana, S. Scherer ”VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015

[7] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao ”3D ShapeNets: A Deep Representation for Volumetric Shapes” in Conference on Computer Vision and Pattern Recognition (CVPR), 2015

[8] H. Su, S. Maji, E. Kalogerakis, EG. Learned-Miller ”Multi-View Convolutional Neural Networks for 3D Shape Recognition” in International Conference on Computer Vision (ICCV), 2015

[9] S. Yu, T. Westfechtel, R. Hamada, K. Ohno, S. Tadokoro ”Vehicle Detection and Localization on Bird’s Eye View Elevation Images Using Convolutional Neural Networks” in IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), 2017

[10] CR. Qi, L. Yi, H. Su, LJ Guibas ”PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” in Advances in Neural Information Processing Systems (NeurIPS), 2017

[11] Y. Wang, Y. Sun, Z. Liu, SE. Sarma, MM. Bronstein, JM. Solomon ”Dynamic Graph CNN for Learning on Point Clouds” in ACM Transactions on Graphics (TOG), 2019 Oct 10;38(5):1-2.

[12] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, M. Bennamoun. ”Deep Learning for 3D Point Clouds: A Survey.” in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020 Jun 29, early access.

[13] A. Serna, B. Marcotegui ”Detection, Segmentation and Classification of 3D Urban Objects Using Mathematical Morphology and Supervised Learning” in ISPRS Journal of Photogrammetry and Remote Sensing, 2014 Jul 1;93:243-55.

[14] X. Roynard, JE. Deschaud, F. Goulette ”Paris-Lille-3D: A Large and High-quality Ground-Truth Urban Point Cloud Dataset for Automatic Segmentation and Classification” in The International Journal of Robotics Research (IJRR), 2018 May;37(6):545-57.

[15] I. Bogoslavskyi, C. Stachniss ”Efficient Online Segmentation for Sparse 3D Laser Scans” in PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science, 2017 Feb 1;85(1):41-52.

[16] A. Nguyen, B. Le ”3D Point Cloud Segmentation: A Survey” in IEEE Conference on Robotics, Automation and Mechatronics (RAM), 2013.

[17] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, ”How Transferable are Features in Deep Neural Networks?” in Advances in Neural Information Processing Systems (NeurIPS), 2014.

[18] K. Simonyan, A. Zisserman ”Very Deep Convolutional Networks for Large-Scale Image Recognition” in International Conference on Learning Representations (ICLR), 2015

[19] F. Chollet ”Xception: Deep Learning with Depthwise Separable Convolutions” in Conference on Computer Vision and Pattern Recognition (CVPR), 2017

[20] N. Mizuno, K. Ohno, R. Hamada, H. Kojima, J. Fujita, H. Amano, T. Westfechtel, T. Suzuki, S. Tadokoro ”Enhanced Path Smoothing Based on Conjugate Gradient Descent for Firefighting Robots in Petrochemical Complexes” in Advanced Robotics (AR), 2019 Jul 18;33(14):687-98.