Towards detecting speed bumps from MLS point clouds data

Hongchao Fan [1]*, Yiping Chen [2]

[1] Department of Civil and Environmental Engineering, Norwegian University of Science and Technology, Trondheim, Norway – hongchao.fan@ntnu.no

[2] School of Geomatic Engineering, Sun Yat-sen University, Zhuhai, China, chenyp79@mail.sysu.edu.cn

ABSTRACT: High definition maps contain a high level of information about the road and its features. This includes traffic signs, speed limits and lane markings, that are not generally included on regular maps. However, they may still lack critical information, such as the locations of speed bumps. Without the information about the whereabouts of speed bumps the passengers' comfort and safety, as well as the condition of the car may be jeopardized. There are currently methods for detecting speed bumps that use changes in acceleration or 3D cameras. However, these approaches are susceptible to external influences interfering with the recorded data. Hence, it is desirable to have this information stored and available through HD maps. In this work, we tested two deep learning-based approaches (PointNet++ and PointCNN) and compared the results with conventional region growing method, in order to find out pros and cons of the modern deep learning-based methods. For our test, we used MLS (mobile laser scanning) point clouds data in Trondheim, Norway.

KEY WORDS: HD maps, point clouds, speed bumps, pointNet, rule-based method

1. INTRODUCTION

Speed bumps are very common in many countries. They are normally set in densely populated urban areas to deaccelerate the speed of vehicles, in order to reduce the risk of traffic accidences, as well as the noise made by vehicles. According to the standard [1] in the road construction defined by the Norwegian Public Road Administration, it is required to mark both sides of speed bumps with reflections (Fig.1a.) and put a warning sign (Fig.1b.) about 20 to 150 meters ahead of the speed bumps depending on the speed limitation on the road.

a. An example of a speed bump marked with reflection on both sides

b. Warning sign of a speed bump

Fig.1. An example of a standard speed bump with warning sign

Unfortunately, due to various reasons, the regulations of speed bumps are rarely obeyed. To the knowledge of the authors of this paper, more than 98% of speed bumps in the City of Trondheim in Norway have neither been marked with reflection materials, nor has any warning sign been put ahead of a speed bump. In fact, the reality in Norway is more crucial than one can expect. Besides the abovementioned situation, many speed bumps have been built with the same material as the road. As one can see in Fig.2., there is a speed bump in the middle of the road. However, it is hardly to see the speed bump even with a short distance, for instance less than 10 meters. The picture was taken in a high position while walking. When sitting in a car for driving, the position is much lower and it is almost impossible to observe the speed bump. Imaging that a car is going with a speed of 30 or 40 km/h on the road, overlooking the speed bump and passing it without speed reduction, there will be then a sudden bumpy in the car and throw the passengers to the ceiling of the car. In such a case, a loud and ear-piercing noise will be heard due to crashing of the car and the speed bump. This makes the drive very uncomfortable and even very danger.

Fig.2. A speed bump in Trondheim, Norway

Therefore, it is necessary to extract the speed bumps and put the information on maps for navigation. There are currently methods for detecting speed bumps that use changes in acceleration [2] or 3D cameras [3]. These methods work only with special conditions, for example, the car for data acquisition has to be driving in a very consistent manner and the speed

* Corresponding author

bumps are very obvious to be seen from the road as background. For situations similar to those as shown in Fig.2., it is only feasible to detect or extract the speed bumps from airborne or mobile laser scanning point clouds.

In this work, we tested two deep learning-based approaches (PointNet++ and PointCNN) to detect speed bumps from mobile laser scanning point clouds. In further, these methods are compared the results with conventional region growing method, in order to find out pros and cons of the modern deep learning-based methods. For our test, we used MLS point clouds data in Trondheim, Norway.

2. DATA USED IN THE EXPERIMENT

The MLS point clouds data used in the experiment was provided by the Trondheim municipality in Norway. The data was collected using a RIEGL VMZ installed on the ceiling of a surveying car on the October 28th, 2018. The European Terrestrial Reference System 1989 (ETRS89) is used to represent the coordinates, and the projection used is the Universal Transverse Mercator (UTM) zone 32N. Normal Null 2000 (NN2000) is utilized as the vertical datum. The dataset adds up to a total of 5899 separate files, where each file is a part of a road, resulting in a dataset of 171GB and a total of 6 578 995 120 points. Each point from the point clouds contain global coordinates, intensity and the time the point was recorded.

The original dataset spans across the entire municipality and includes both rural and urban ares. The extent of the provided point cloud files is shown in Fig.3., while Fig.4. shows the bounding boxes of all the LiDAR files, indicating the density of data points around the municipality. The different roads vary from dirt roads with holes and vegetation, to clean asphalt roads with minimal noise and anomalies. The area's topography also varies, from relatively flat areas to more steep terrain. Due to bumps more commonly being found within cities and more densely populated areas, most of the study area for annotation was in or near the city center.

Fig.3. Overview of the extent of MLS point clouds

To reduce the computation cost and the complexity of the data analysis, 3D point clouds of road surface need to be extracted at first. For this purpose, Zhang [4] extracted road surface point by using the elevation information from the range data. In 2013, Hervieu and Soheilian [5] computed angular distance to ground normal map, and detected road edges by applying prediction on this feature map to detect road edges, which were further used to extract road surface points. In 2013, Ishikawa et al. [6] organized data into vertically segmented gridded form and height, and extracted road surface points by looking at the variance of height direction.

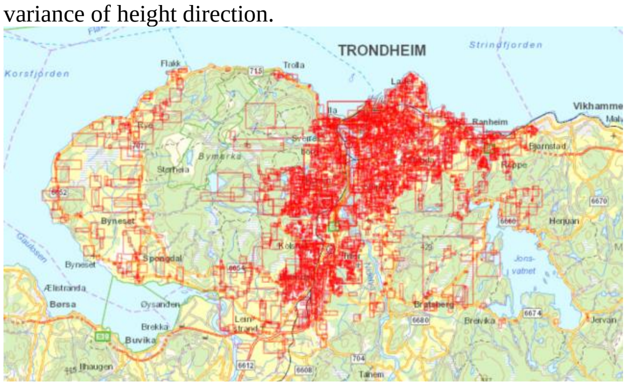


Fig.4. Bounding boxes of all LiDAR files

In this work, we used a relatively simple method of overlapping 2D vector data with 3D point clouds because the positional accuracy of the collected data is very high. In Norway, FKB (Common Map Database) contain road data, including accurate polygons of the whole road network, see Fig.5. This data is available at the online service [7].

Fig.5. Road polygon overlapped with aerial image

The 3D point clouds acquired by the mobile laser scanner cover not only road surfaces, but also trees, buildings and other urban objects along roads, as illustrated in Fig.6.

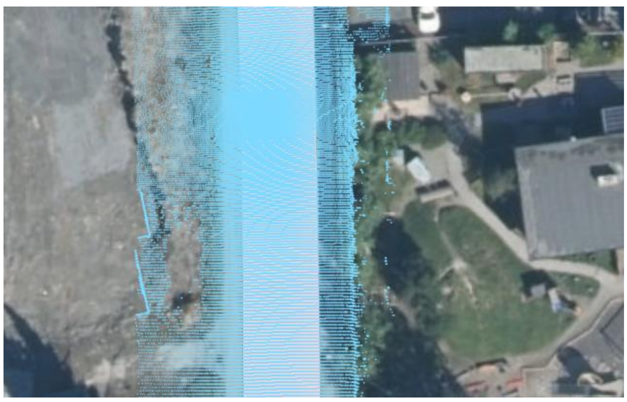


Fig.6. MLS point clouds overlapped with aerial image

In this work, we overlapped the road polygons with 3D MLS point clouds and keep all the points with the road polygons as 3D points on road surface, as depicted in Fig.7. It should be noted that the road polygons might not cover all the road in the width direction due to the positional accuracy of the RTK measurement for the MLS data collection. But this does not affect the extraction of speed bumps because speed bumps are normally large enough to be extracted.

Fig.7. clipped 3D point clouds on road surface

After the point clouds on road surface are separated from the original MLS point clouds, a training data set of speed bumps for deep learning is established. In this work, four different types of speed bumps are distinguished and labelled in the 3D MLS point clouds. These four types of speed bumps are circular bump (Fig.8.), modified circular bump (Fig.9.), trapezoidal bump (Fig.10.) and speed cushion (Fig.11.).

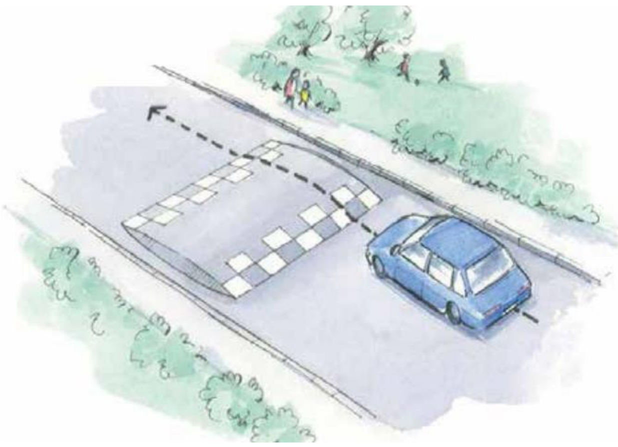


Fig.8. Circular bump

Fig.9. Modified circular bump

Circular bump has the shape of a circle segment, and is the simplest and most common type of bump. Modified circular bump is a circular bump with counter curves at the ends to give a softer start and end of the bump. This gives less discomfort than circular bump, due to blows to the wheels. Trapezoidal bumps are designed with a flat top surface and sloping, flat surfaces on at the start and end of the bump. It is suitable where you need to establish a flat surface on top of the bump, for example at an elevated walkway. Speed cushions only covers a part of the road lane, and consist of a flat square top and flat ramps on the sides. They are designed so that vehicles with small wheelbase, i.e passenger cars, must pass with at least one wheel on the pad. These are common in areas with bus traffic, as they affect the accessibility of the buses to a lesser extent as they can pass the bump without too much speed reduction, thus strengthening public transport's competitiveness compared with passenger cars. On two-lane roads, speed cushions are laid pairwise next to each other.

Fig.10. Trapezoidal bump

Fig.11. Speed cushions

When choosing an annotation tool, there are several things the tool should be capable of. Firstly, it should be able to handle large amounts of data, as point cloud datasets can often contain billions of points. Secondly, it should have an easy-touse interface, ideally with the possibility to visualize multiple viewpoints (side, top, front) at the same time. It should preferably be online, as this makes it much easier to divide the annotation jobs. A tool that fulfils all these requirements is Supervise.ly [8].

Using the software, the 3D bounding boxes are drawn as tightly as possible around the speed bumps, and they should capture all visible parts of the bump. If a particular bump is bad due to bad maintenance of the road or wear-and-tear, the bounding box is created on a best-effort basis. During the annotation of the dataset, we encountered several abnormal bumps that should be addressed, namely, 3D point clouds of some speed bumps are partly missing (see Fig.12.) due to various reasons. These kind of samples are also taken into the training data set, so that the deep learning process could learn better features of speed bumps and thus make good prediction.

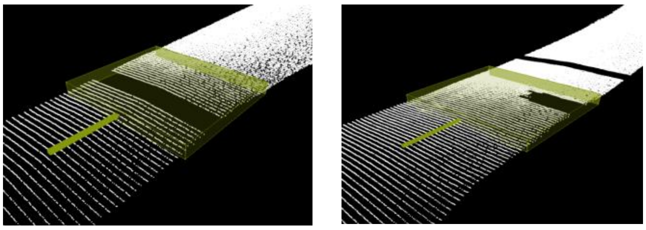


Fig.12. partly missing speed bumps as samples in training data

As shown in Table 1., we totally annotated 921 speed bumps from the MLS point clouds in Trondheim. Among others, 584 speed breakers were annotated, whereas 17 of these were considered as bad bumps, 42 had missing parts and 9 had missing line segments. The number of different speed cushions were 337, of which 5 of these were considered as bad bumps, 9 of them had a missing part, and none of them had missing line segments. It makes sense that number of speed breakers with missing parts and missing lines are greater than that of speed cushions, as speed breakers are much larger than the counterpart, making the probability of the occurrence of such phenomenon greater.

Table 1. statistic overview of training data

	Total	Bad bumps	Missing part	Missing Line Segments
Speed breakers	584	17	42	9
Speed cushions	337	5	9	0
Sum	921	22	51	9

3. EXTRACTION OF SPEED BUMPS

The task of locating speed bumps from a derived point cloud, can be defined as a part segmentation task, where each bump is a separate part of a road. Two networks, namely PointNet++ [9] and PointCNN [10] were chosen as deep neural networks to evaluate the suitability of the proposed dataset.

Two minor implementations was appended to both PointNet++ and PointCNN. This included calculating class weights and implementing early stopping. As the proposed dataset is unbalanced, we calculated class weights by analyzing the number of points in the different classes in the training set. This weight was sent to each of the loss functions in an attempt to minimize the class imbalance issue. In order to stop the model from overfitting on the training data, early stopping was implemented. The loss function was used as a condition for when the model should stop the learning process. The early stopping was performed when the validation loss did not decrease further, after 7 epochs. For both of the network, the optimizer of ("Adam") was used. In terms of loss function, NLLLoss was used for PointNet++ and Sparse softmax cross entropy was used for PointCNN.

To draw a conclusion if the deep learning based approaches are better than the conventional method, we also used region growing approach to detect speed bumps. The method calculates normal vectors in a small window that needs to be shifted along a road.

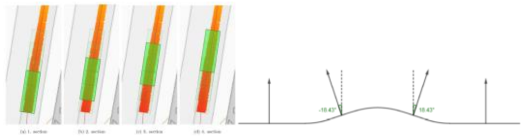


Fig. 13. Region growing method for detection of speed bumps

As denoted in Fig.13., a window is moving along the road in point clouds. For each movement, the normal vector within the window will be calculated. In case there is a speed bump, the normal vector will change from the beginning to the end of the speed bump.

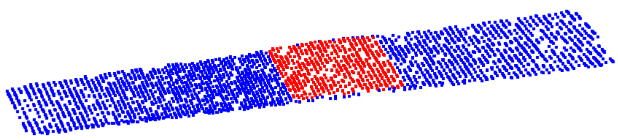
4. EXPERIMENTAL RESULTS

The results of extracting speed bumps using PointNet++ and PointCNN are presented in Table 2. These results are based on the classes of individual points, checking each point. To the overwhelming amount of road points the values for classifying road points is generally very good. Looking at class averages PointCNN outperforms PointNet++ if looking at the recall, while the opposite is true for class average of IoU.

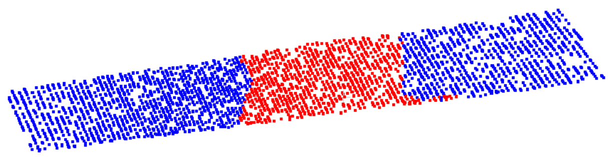
Table 2. Results of speed bump extraction using deep learning

Model	Type	Recall	IoU
PointNet++	Road	0.969	0.943
	Speed bump	0.830	0.695
	Speed cushion	0.835	0.824
	Class average	0.878	0.821
PointCNN	Road	0.982	0.972
	Speed bump	0.901	0.775
	Speed cushion	0.839	0.649
	Class average	0.907	0.799

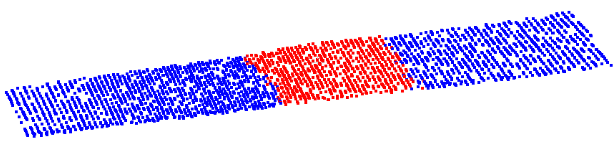
In Table 2, one can see that the extraction results are differently when checking the 3D points belonging to individual speed bumps. However, when checking the exact points extracted from the 3D points on the road surface, one can see that although there is difference in the extraction, the most of points are identical. Most importantly, the positional accuracy of the detected speed bumps is almost same, as shown in Fig 14.



a. Ground truth



b. Extraction result by PointNet++



a. Extraction result by PointCNN

Fig.14. Speed breaker as predicted by the two models

As for speed cushion, the deep learning-based method delivered good results, as demonstrated by an example in Fig.15.

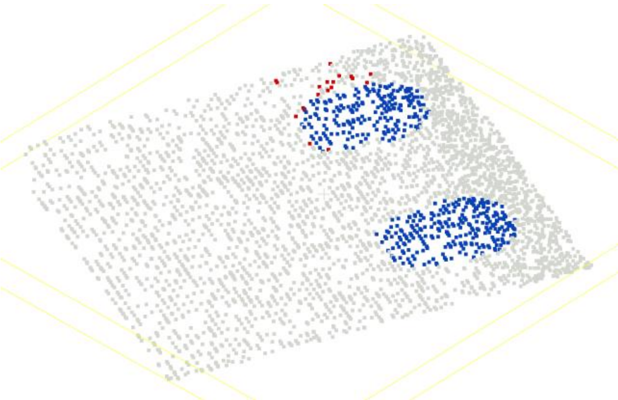


Fig.15. extraction results of a speed cushion (grey points as road surface, blue points as speed bumps and red points as noise)

As mentioned previously, the method of region growing was used to check if deep learning based methods have advantages comparing to the conventional method. In Table 3., the results of extracting speed bumps using region growing method is listed.

Table 3. Results of speed bump extraction using region growing

Type	Recall	IoU
Road	0.959	0.923
Speed bump	0.851	0.665
Speed cushion	0.866	0.873
Class average	0.892	0.820

The experimental results shows that deep learning-based methods achieve similar results as the conventional region growing method. There is one obvious advantage of the region growing method, namely, it does not need training data and therefore could be applied anywhere.

5. CONCLUSION

In this paper, a comparison study is presented to investigate whether the most updated AI methods (PointNet++ and PointCNN) have obvious advantages for extracting speed bumps than the conventional method of region growing. For this purpose, training data set was established using MLS point clouds data in Trondheim, Norway.

In the original plan, four different types of speed bumps (circular bump, modified circular bump, trapezoidal bump and speed cushion) were expected to be extracted and distinguished from point clouds on road surface. Unfortunately, the circular bumps, modified circular bumps, and trapezoidal bumps cannot be clearly identified and distinguished in the prediction process. For this reason, these three types of speed bumps are merged as one big type in the training data set later. Similarly, this cannot be successfully done by using region growing method, either. The reason is that these three types of speed bumps are constructed very similar and with different width and height in the reality.

The experimental results showed that both deep learning methods and conventional method achieved good results. Although point coverages of individual speed bumps are different from each other which makes the statistic in the results look differently, the positional accuracy of the extraction results is almost same, since speed bumps are very large and even part of them are segmented differently their location is identical.

In terms of efficiency, deep learning-based methods are better than the conventional region growing method. But the region growing method does not need training data.

We did not test the performance of extracting speed bumps from point clouds with different point densities. But we are very sure that point density matters. It would greatly affect the extraction results. When comparing deep learning-based methods and the conventional region growing method, we think that the influence of point density on region growing method should be less than the deep learning-based methods.

In the future, the training data set will be improved with more samples of the two types of speed bumps (speed breaker and speed cushion). In addition, the comparison will be conducted by generating point clouds and sample data with different point density, in order to check the influence of point density on deep learning.

ACKNOLEDGEMENT

The experiments presented in this paper were carried out in line with a master thesis by Sverre Kjuus and Lars Eivind R Lien at the Norwegian University of Science and Technology in 2022.

REFERENCES

[1] NPRA Directorate, 2014, Work on and along roads: Requirements and guidelines regarding warning and protection. Handbook N301E. https://www.vegvesen.no/globalassets/fag/handboker/hb-n301e.pdf (accessed on the July 15th, 2023)

[2] Jose M Celaya-Padilla et al. 2018. "Speed bump detection using accelerometric features: A genetic algorithm approach". In: Sensors 18.2 (2018), p. 443.

[3] Kwang Ming Lion, Kae Hsiang Kwong, andWeng Kin Lai. "Smart speed bump detection and estimation with kinect". In: 2018 4th International Conference on Control, Automation and Robotics (ICCAR). IEEE. 2018, pp. 465–469.

[4] Wende Zhang. "Lidar-based road and road-edge detection". In: 2010 IEEE Intelligent Vehicles Symposium. IEEE. 2010, pp. 845–848.

[5] Alexandre Hervieu and Bahman Soheilian. "Road side detection and reconstruction using LIDAR sensor". In: 2013 IEEE Intelligent Vehicles Symposium (IV). IEEE. 2013, pp. 1247–1252.

[6] Kiichiro Ishikawa et al. "Recognition of road objects from 3D mobile mapping data". In: International Journal of Cad/CAM 13.2 (2013).

[7] FKB data: https://geonorge.no

[8] 2019. url: https://supervise.ly/.

[9] Charles R Qi et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space". In: Advances in neural information processing systems 30 (2017).

[10] Yangyan Li et al. "PointCNN". In: CoRR abs/1801.07791 (2018). arXiv: 1801.07791. url: http://arxiv.org/abs/1801.07791.