PointAtMe: Efficient 3D Point Cloud Labeling in Virtual Reality

Florian Wirth[1], Jannik Quehl[1], Jeffrey Ota[2] and Christoph Stiller[1]

***Abstract*— Generating annotations which can be used to train new models has become an independent field of research within machine learning. Its goal is producing highly accurate annotations as cost efficient as possible. 3D point clouds are the common sensor output when recording 3D data from a mobile platform. The latest ways of annotating 3D point clouds include their visualization on a 2D screen. This method contradicts the goal of time-efficient annotating since it is unintuitive and therefore unnecessarily time consuming. We present a novel labeling technique in Virtual Reality. Using our tool, we accelerate the process of data annotation significantly compared to existing approaches. Furthermore, we will give the machine learning community access to our tool and create a new community-labeled dataset for autonomous driving. Furthermore we plan to set up an annotation benchmark in which primarily commercial annotation companies but also researchers active in annotation can take part in. We present results from an experimental plattform based on Oculus Rift indicating a huge potential for VR annotations.**

***Index Terms* — label tool, dataset, benchmark, machine learning.**

I. Introduction

Annotating data in a fast and consistent way has recently become an independent field of research. An increasing number of companies chooses the manifacturing of huge amounts of data annotations in a cheap way as their business model. In machine vision, this trend started with annotating images. With increasing quality and point density at a decreasing cost of LiDAR sensors, especially in the field of automated mobile systems annotating in 3D point clouds has become important for data annotators.

Meanwhile, in the field of consumer electronics and gaming, tools for visualizing three-dimensional environments in its native way are getting more popular. With a large amount of cheap or free software tools and games for these Virtual Reality (VR) devices – such as the game engine Unity[1] – the VR community, its tools, tutorials and forum contributions, are permanently growing. Therefore, it is getting easier for researchers not active in game designing, game development or VR applications to benefit from this development.

The automated driving community, however, is still annotating 3D point clouds on 2D computer screens with 2D mice or touch screens, thus sacraficing a whole dimension due to hardware. Therefore, we try to leverage the development in game design and the natural human way of perceiving its environment in 3D. We developed a VR label tool which allows annotating point clouds in their natural habitat: A 3D representation projected around the human annotator.

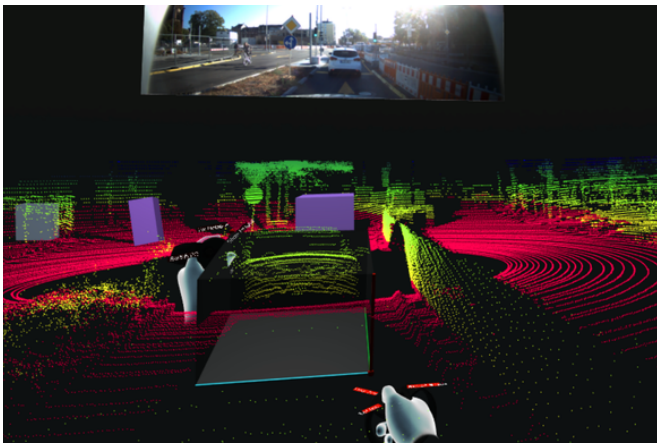


Fig. 1: Scene recorded from a mobile plattform, labeled exemplarily.

According to our experiments, people not participating in the development of this tool are able to learn annotating data quickly after receiving a brief introduction in tool handling. Our tool accelerates the process of data annotation by a factor of three compared to the latest publication in this field. Experiments with KITTI [1] labels show the benefit in annotation accuracy our method offers. During data annotation for several private and public datasets at our institute [2] we observed a fast decreasing work motivation of student assistants not working with their annotations themselves afterwards. Our tool tackles this problem by gamifying data annotation and we would like to motivate the community working actively with Ground Truth themselves to contribute to our project. We open-sourced the label tool and some post processing scripts on github[3] and in the Unity AssetStore[4]. Furthermore, our tool is the only known tool to set 3D bounding boxes with 9 degrees of freedom (DoF). Two more DoF than usual are especially useful for objects on tilted roads, like in [2] Fig. 6: On the right side cars park at a steep road in San Francisco, but the bounding boxes are parallel to the ego vehicle.

II. Related Work

The state of the art in annotating 3D data is to visualize a point cloud on a 2D screen and choose a subset of points by

*This work was supported by Intel Corporation

[1]are with Intitute of Measurement and Control Systems, Karslruhe Institute of Technology, Karlsruhe, Germany {firstname.surname}@kit.edu

[2] Jeffrey Ota is with Intel Corporation, Santa Clara, CA, U.S.A. jeff.m.ota@intel.com

[1]Unity Technologies, www.unity3d.com

[2]https://www.mrt.kit.edu/software/datasets.html

[3]https://github.com/florianwirth/PointAtMe

[4]https://assetstore.unity.com/

drawing a hull around them. This process is often supported by camera images taken in the same scene. By making use of extrinsic calibration, the 3D annotation can be projected into the image. Antsy 3D is a browser based tool for point-wise annotations[5] without related conference contribution. Russel *et al.* [3] extended their online labeling tool „LabelMe“[6] with a toolbox for Matlab[7].

Wong *et al.* [4] proposed a learning based annotating tool which proposes cuboid candidates in RGB-D images given e.g. by Microsoft Kinect. According to the authors, this tool is meant for annotating indoor scenarios only. Xiao *et al.* [5] proposed a browser based open-sourced annotation tool to annotate RGB-D images. By annotating the same object in consecutive frames with different camera positions the objects 3D shape can be reconstructed. It is presented in indoor scenarios only but it can also be used outdoors. However, this procedure can be used within static scenarios only.

Veit *et al.* [6] published a framework to annotate points in 3D using Unity3D. However, they stated the superior manipulation of 3D data using a 2D interface so they tracked a smartphone with 6 DoF for having a large variaty of input possibilities, such as writing text or choosing cuboids on a touch screen. Wilkes *et al.* [7] observed that tracked multi-touch mobile devices do not lead to advances in performence when used in VR. Furthermore, they suggest to use a 6 DoF device visualized in VR or use VR buttons and sliders to replace the smartphone functionality. Yu *et al.* [8] developed algorithms for spatial, structure-aware selection method that allow users to draw a lasso around a 2D projection of characteristic structures within a point cloud. However, this method is computationally expensive and therefore ineffective for setting up large databases as machine learning requires them at the moment. Coffey *et al.* [9] presented a VR system which is making use of two displays for navigation. Multi gestures on the touch screen allow interfering with the object while the user only sees the 3D object in VR. Spectators can follow the users presentation on the second, larger screen. This tool was developed in the field of medical visualization of internal organs and isn't tailored to the problem of annotating convex objects which most traffic participants can be reduced to. Bacim *et al.* [10] and Lubos *et al.* [11] present two methods to annotate point clouds with free-hand gestures. Their accuracies may be higher for the application of segmenting complex 3D point clouds. Since we are interested in geometrically simple objects, their methods would unnecessarily increase the annotation effort and cost. Monica *et al.* [12] developed a control point based tool. It allows the user to set several control points around a single instance within a point cloud. A segmentation algorithm then provides real time feedback for the annotator who can then provide an improved cluster of control points. [12] are the only ones conscientiously evaluating their label tool with regard to annotation speed. They use a 2D rectangle-based method as baseline. Therefore, we try to compare our approach with theirs.

[5]https://github.com/alvinwan/antsy3d

[6]http://labelme.csail.mit.edu/Release3.0/

[7]LabelMe3D, http://labelme.csail.mit.edu/Release3.0/

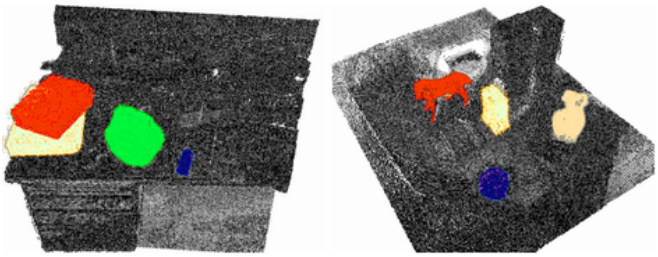


Fig. 2: Monica *et al.* [12] propose a tool to annotate point clouds pointwise. They evaluate their tool by measuring the time needed for unexperienced annotators to annotate a point cloud. We try to compare ourselves with their results, even though their scene is not publicly available.

There are already several datasets which include 3D annotations of road users. The KITTI dataset [1] includes 80,256 annotated objects in about 15,000 LiDAR scans. Nuscenes [13] is the largest known dataset at the moment. It contains around 1.1 bounding boxes for traffic participants in 400,000 LiDAR sweeps recorded from a Velodyne HDL 32. H3D dataset [2] contains 1.0 million objects, that are labeled in 27,000 Velodyne HDL 64 sweeps, so their scenes are more crowded. Furthermore, the TUBS Road User Dataset [14] was announced for early 2019. It contains 12,000 semi-automatic labeled scans from a Velodyne HDL 64.

Even though there are datasets which already tackle the problem of 3D object detection/tracking by making use of Ground Truth generated from high resolution point clouds, the requirement for data by emerging deep learning techniques demands for getting access to more data from different sensors in different environments and different traffic situations.

III. 3D Labeling in Virtual Reality

We created a tool for annotating 3D point cloud data in 3D by making use of the Virtual Reality device Oculus Rift and its Touch Controllers. The data can be sequence based so it is possible to annotate a whole sequence in one run and keep track of the same object over time. The tool is supposed to be easy to use and simple to modify for the individual needs of research groups that would like to benefit from our work. We introduced some common parameters which can be set easily for the individual needs dependent of the sensor setup used.

A. Concept

During the whole process of data annotation, we see a transparent dummy bounding box between our controllers. This box has 9 DoF: Its position and scale is defined by the red anchors mounted directly to the virtual controllers. The orientation of the box is determined by the orientation of the right controller only. To make the annotator aware of this at any time, we mounted a coloured set of axes to

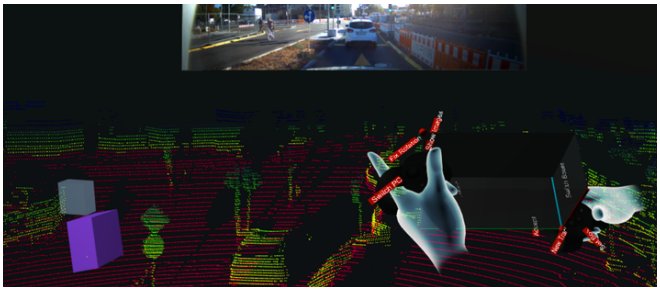


Fig. 3: Tool environment: Point cloud and images are visualized. Two cyclists are labeled. Dummy bounding box between between controllers shall be adopted to the shape of an object.

the right controller. It is the goal of our annotation tool to make the shape of the box represent the real world object as well as possible. Therefore, the point cloud and the images from the measurement run of a stationary or mobile plattform is visualized in Unity (Fig. 1 and 3). In many cases, the shape has to be guessed for the most part. We will add a postprocessing script which improves the boxes. The orientation has to be set manually as precise as possible. When annotating data, all points belonging to one object have to be inside the dummy box, all points not belonging to the object have to be outside. We tried to support the annotator on that task by making the back wall of the box black and intransparent and the front of it dark but transparent. All points of the object have to be visible (in front of the back wall) but slightly dimmed (behind the front wall). The tool outputs a text file with a list of objects for each point cloud. Position, scale, orientation (as a quaternion) relative to the LIDAR and meta information are exported for each object.

B. Controller Key Assignment

We strictly separated the functionality of both Oculus Touch controllers to make the annotation process as intuitive as possible. The left controller contains functionality related to scene understanding and scene managing only. The right controller only contains functions directly related with annotation. Functionality is assigned to buttons and joystick as follows:

1) **Left Oculus Touch Controller**
   a) **Hand trigger: Grab PCL** Press to move the point cloud with 6 DoF. We accelerated the translational movement, otherwise the annotation process would be physically demanding after a short time span.
   b) **Index trigger: Fix Rotation** Often, the bounding box does not require all of the 9 DoF to be released. The index trigger can be pressed to release two rotational DoF both of the box and of the point cloud.
   c) **Thumbstick: Switch PCLs** Since the label tool is meant to work sequence based, the thumbstick allows to switch scenes.

Fig. 4: 3D visualization of the key assignment. Keys on the left controller are used for scene understanding, keys of the right controller are used for data annotation.

   d) **Button Y: Show Images** To get a good overview of the current scene four images can be shown, one for each direction.
   e) **Button X: Switch Scale** To comfortably annotate objects of different scales the scene's scale can be changed.
2) **Right Oculus Touch Controller**
   a) **Index trigger: Accept** Meta data is added by dialogs which pop up when a new track or a new box is created. These dialogs can be accepted with the index trigger.
   b) **Thumbstick: Switch Tracks** The track the annotator currently works on can be changed by pushing the thumbstick to the left or to the right.
   c) **Button A: New Track** Create a new track by hitting A. The „new track"- dialog will pop up and meta information can be added.
   d) **Button B: New Box** By hitting B, a new bounding box for the current scene is created within the current track. The „new box"- dialog will pop up and information about the label quality can be added.

C. Meta Information

Besides knowing the precise location of a road user it is important to understand its current role in the traffic scene and the resulting relation to the ego vehicle. To tackle this problem we add meta information to each object which is queried within dialogs that pop up everytime a new track gets started. We, for a start, added five properties, which are the most important for our needs:

1) **Type of Road User**: We distinguish between five types of road users. One class contains road users that have the same physical model.
   a) **Pedestrian**: Slow vulnerable road user (VRU). It can change its velocity and its direction of movement anytime and uses primarily its own road infrastructure. (pedestrian, wheelchair user, pedestrian with buggy, child)

b) **Two Wheeler**: Fast VRU. It steers by weight transfer and can therefore not change velocity and direction of movement independently. Both DoF of the bicycle model used for four-wheeled vehicles introduced by Taheri *et al.* [15] are connected by a non-holonome contraint leading to the model presented by Neimark *et al.* [16]. It shares the road with the ego vehicle but has the spatial dimensions of a pedestrian rather than of a car. (cyclist, motorbike, segway PT, scooter, motorized wheel chair)

c) **Car**: Same physical model like to ego vehicle [15]. (everything with three or more wheels, including trucks)

d) **Train**: Vehicle with only one DoF, its velocity. (train, tram, subway)

e) **Trailer**: Moveable object strictly following another moveable object. Only body shall be within a box, not the trailer hitch. (trailer(s), rear wagon(s) of a train, rear part(s) of an articulated bus)

2) **Priority**: Does the road user have priority with respect to us if dynamic road signs (traffic signs, traffic policemen) are ignored? Only applies to road user crossing our road. Otherwise the priority is indifferent.
3) **Direction**: Does the road user drive in our direction or in the opposite direction? Crossing road user shall be labeled as indifferent.
4) **Lane**: Lane on which the traffic participant is currently located relative to the ego vehicle's lane. We consider more than ± 2 lanes to be not directly relevant for the ego vehicle.
5) **Participating in Traffic**: Is the vehicle participating in traffic or is it parking?

A more detailed label instruction is given online.

IV. Results and Evaluation

When annotating data in a large scale for a dataset, two characteristic numbers related to the method of data annotation matter: The annotation time since it is proportional to the annotation cost and the annotation accuracy. The annotation cost shall be minimized, the annotation accuracy shall be maximised.

To the best of our knowledge there is no benchmark for data annotation to compare ourselves with. As the only reference [12] evaluated their tool in a conscientious way. They annotate data point wise, however, the objects they use indeed can be assumed to be cuboid-like. Except for the toy horse and the vase the concave parts of the objects they use are negligible as much as the concave parts of a traffic participant. [12] did not publish their evaluation data. We try to mimic their evaluation method as precisely as necessary to roughly make our results comparable to theirs.

Monica *et al.*[12] chose two scenerios each with four objects which shall be labeled pointwise. They use unstructured indoor scenes shown in Fig. 2. The first scene is a messy workbench with around 20 different objects. The second scenario is a plate with around 10 objects standing upright next to each other. Only 4 objects (colorized in Fig. 2) of each scene are supposed to be labeled. Monica *et al.*measure the time the annotators need to annotate the scenes with two techniques. The first one is their proposed control point based method. The second method is based on selecting rectangles. The annotator chooses a rectangle which contains all points for the current direction of view. It is applied from several viewing directions. The average result of 10 annotators is given in Table 2. and 3. in [12].

For our use case, the annotation speed can be evaluated easily by measuring the annotation time of unexperienced annotators. We did so by making use of four KITTI 3D Object Detection scenes. We asked students in one of our courses to annotate data. We offered an automated ride with our experimental vehicle Bertha [17] as reward for the three best results compromized between annotation speed and accuracy. The students hadn't used the tool before. They had 30-45 min for getting familiar with the tool while we gave them some hints how to label objects efficiently. Afterwards they labeled four KITTI training scenes which they had not seen before.

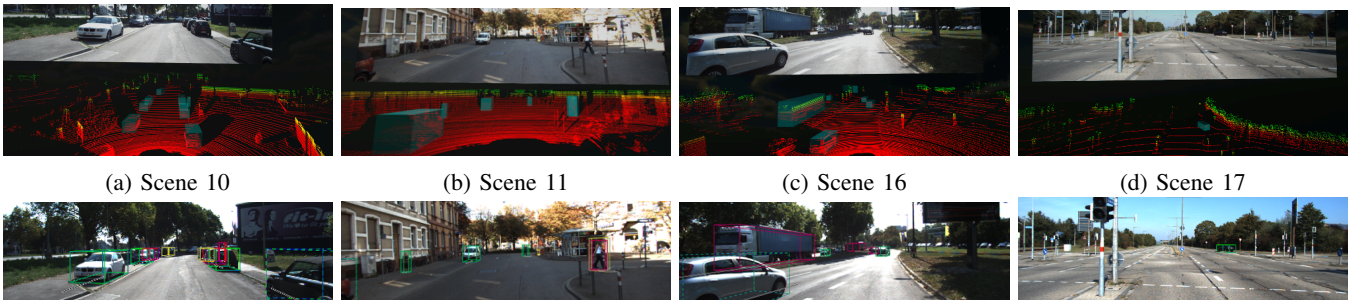
The accuracy is rather difficult to evaluate. As the images of the KITTI 3D bounding boxes show in Fig. 5e, the labels differ from a box an annotator would choose in VR. For example, most boxes in the distance are below ground level (images in third row) which can also be observed when looking at the images provided within the KITTI development kit (forth row). Therefore, we use KITTI scenes for evaluation but generate our own Ground Truth (GT). We, the developers of this tool, do already have a lot of experience in annotating in VR, so it is likely that we produce more accurate annotations than our student annotators. Labeling those four GT scenes took us 37.5 seconds per object. For our students, we specified three requirements: All points of the current object are supposed to be within the box, all points not part of the current object shall be outside the box, and the shape of the box shall match the shape of the physical object as good as possible. To assess the label accuracy, we count the relative number of False Positive (FP) and False Negative (FN) points per object, as shown in table I.

TABLE I: Results compared with KITTI and our GT.

FP / FN ratio per Object	PointAtMe (GT)	Experiments	KITTI
PointAtMe (GT)	0% / 0%	**7.2% / 3.3%**	17.2% / 19.7%
Experiments	⋱	—% / —%	**30.7% / 13.7%**
KITTI	⋱	⋱	0% / 0%

Note that our students' results are way closer to our generated GT than to the GT offered by KITTI which indicates the commensurability of relabeling KITTI data for our needs. The comparatively high FP of our students is due to objects which were hit by a few points only.

We can now roughly compare our results with Monica *et al.* [12]. They evaluated the average task completion time,



(a) Scene 10 (b) Scene 11 (c) Scene 16 (d) Scene 17

(e) Corresponding images and 3D bounding boxes created with KITTI dev kit [1].

(f) Ground line enters bounding box. (g) Bounding boxes to tight, points hitting the object are outside the box. (h) Box of truck does not fit. (i) Ground line enters bounding box.

(j) Detailed view of KITTI 3D Ground Truth boxes in KITTI dev kit. Especially boxes in the distance do not match with the ground plane or the real size of the vehicle according to the point cloud. Points outside the box are bright, points inside are dimmed.

Fig. 5: KITTI Ground Truth visualized both in Unity (first and third row) and with KITTI development kit (second and fourth row). The Ground Truth accuracy is not precise enough to evaluate our label tool with it (as is shown in the third and fourth row). In conclusion, we labeled these four scenes ourselves.

the number of undo operations per minute, and the number of annotation errors. One annotation error is one wrong label assigned to a point. Their unstructured point clouds consist of 150,868 and 124,966 point which roughly corresponds to the point clouds used in KITTI (recorded with one Velodyne HDL 64 S2: ~140,000 points / scan). Stats for 9 annotators can be found in the table below.

TABLE II: Results compared with baseline.

Method	**PointAtMe (ours)**	**control points** [12]
Time/Object (sec.)	55.2 ± 12.1	96.0 ± 22.5
Undo/Object.	0.42 ± 0.43	2.1 ± 1.2
Errors (Points/Object)	28.7 ± 5.1	266.1 ± 96.2
FP ratio / Object	7.16%	unknown
FN ratio / Object	3.27%	unknown

To compare the results, we refer to a single object (Monica *et al.* used 8 in total, we used 20). The time necessary to label one object decreases approximately by a factor of two compared to the baseline. The amount of errors per Object is not directly comparable since Monica *et al.* used artificial scenes and we used real world scans. Some objects of ours only where hit by a few dozens of points. Therefore, we added the FP and FN ratio per object to show that on average, our students added 7.16% more points than our GT contains. Also, they left out 3.27% of the points our GT contains.

Our tool promises to generate an interesting new dataset that can be produced in a cheaper and more precise way than before. Furthermore, we can tackle the problem of tired labeling students because labeling data in VR gamifies the annotation process. Still, about a third of the students reported VR sickness or comparable phenomena after the experiments, so one needs to examine applicants for this issue before hiring them for annotating in VR for a long period of time.

V. Conclusions and Future Work

We presented a new label tool for annotating cuboid-like objects in point clouds. Our primary application is 3D data annotation in the environment of mobile robots like they occur in the context of automated driving. We contribute to the community by making our tool publicly available and open-sourced. Furthermore, we announce a benchmark for automated driving which shall be based on community data annotation. We do not only want to benchmark methods in the field of automated driving but also the process of data annotation itself. Our tool outperformes SOTA solutions with regards to annotation speed. Furthermore, we showed the superiority of annotating 3D data in 3D compared to existing

solutions. We did so by visualizing Ground Truth data used within the KITTI 3D Object Detection training set.

Our next steps will be to directly visualize points within a placed bounding box. This is expected to improve the label quality. Furthermore, we will publish raw data for everyone willing to participate in our community-labeled dataset. We experienced a huge benefit when forcing people who have to work with those labels afterwards to annotate data themselves. They label data in a more conscientious way than paid people who never use the data nor the labels ever again. However, we did not evaluate the problem of working in VR for a long time span, yet.

ACKNOWLEDGEMENTS

We would like to thank Intel Corporation for supporting our work.

Our tool was inspired by the work and code of Gerard Llorach[8] which was funded by the IMPART project[9] under the ICT - 7th Framework Program (FP7) from the European Comission. His work was published within the Unity Asset Store (Point Cloud Free Viewer)[10].

Furthermore, we would like to thank Christoph Niewöhner from DTC GmbH Navigation & Security Solutions for helping us with the experimental vehicle we plan to use for our new dataset.

REFERENCES

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[2] H. G. Abhishek Patil Srikanth Malla and Y.-T. Chen, "The h3d dataset for full-surround 3d multi-object detection andtracking in crowded urban scenes," *International Conference on Robotics and Automation*, 2018.

[3] B. C. Russell and A. Torralba, "Building a Database of 3D Scenes from User Annotations," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[4] Y.-S. Wong, H.-K. Chu, and N. J. Mitra, "SmartAnnotator an interactive tool for annotating indoor RGBD images," *Computer Graphics Forum*, 2015.

[5] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A Database of Big Spaces Reconstructed using SfM and Object Labels," *IEEE International Conference on Computer Vision*, 2013.

[6] M. Veit and A. Capobianco, "Go'Then'Tag: A 3-D point cloud annotation technique," *IEEE Symposium on 3D User Interfaces (3DUI)*, 2014.

[7] C. B. Wilkes, D. Tilden, and D. A. Bowman, "3D User Interfaces Using Tracked Multi-touch Mobile Devices," *Joint Virtual Reality Conference of ICAT - EGVE - EuroVR*, 2012.

[8] L. Yu, K. Efstathiou, P. Isenberg, and I. Tobias, "Efficient Structure-Aware Selection Techniques for 3D Point Cloud Visualizations with 2DoF Input," *IEEE Transactions on Visualization and Computer Graphics*, 2012.

[9] D. Coffey, N. Malbraaten, T. B. Le, I. Borazjani, F. Sotiropoulos, A. G. Erdman, *et al.*, "Interactive Slice WIM: Navigating and Interrogating Volume Data Sets Using a Multisurface, Multitouch VR Interface," *IEEE Transactions on Visualization and Computer Graphics*, 2012.

[10] F. Bacim, M. Nabiyouni, and D. Bowman, "Slice-n-Swipe: A Free-Hand Gesture User Interface for 3D Point Cloud Annotation," *IEEE Transactions on Visualization and Computer Graphics*, 2012.

[11] P. Lubos, R. Beimler, M. Lammers, and F. Steinicke, "Touching the Cloud: Bimanual Annotation of Immersive Point Clouds," *IEEE Symposium on 3D User Interfaces (3DUI)*, 2014.

[12] R. Monica, J. Aleotti, M. Zillich, and M. Vincze, "Multi-Label Point Cloud Annotation by Selection of Sparse Control Points," *International Conference on 3D Vision (3DV)*, 2017.

[13] H. Caesar, O. Beijbom, V. Bankiti, A. Lang, S. Vora, and C. Dicle, "NuScenes," *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[14] C. Plachetka, J. Rieken, and M. Maurer, "The TUBS Road User Dataset: A New LiDAR Dataset and its Application to CNN-based Road User Classification," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Maui, USA: IEEE, 2018.

[15] S. Taheri, *An Investigation and Design of Slip Control Braking Systems Integrated with Four Wheel Steering*.

[16] J. I. Neimark and N. A. Fufaev, *Dynamics of Nonholonomic Systems*. American Mathematical Soc., 1974.

[17] S. Tas, N. O. Salscheider, F. Poggenhans, S. Wirges, C. Bandera, M. R. Zofka, *et al.*, "Making Bertha Cooperate - Team AnnieWAY's Entry to the 2016 Grand Cooperative Driving Challenge.," *IEEE Trans. Intell. Transp. Syst.*, 2017.

[8] https://gerardllorach.weebly.com/

[9] https://impart.upf.edu/

[10] https://assetstore.unity.com/packages/tools/utilities/point-cloud-free-viewer-19811