IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 28, 2021, accepted February 22, 2021, date of publication February 26, 2021, date of current version March 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3062547

Annotation Tool and Urban Dataset for 3D Point Cloud Semantic Segmentation

MUHAMMAD IBRAHIM, NAVEED AKHTAR, MICHAEL WISE, AND AJMAL MIAN, (Senior Member, IEEE)

Department of Computer Science and Software Engineering, The University of Western Australia, Crawley, WA 6009, Australia

Corresponding author: Muhammad Ibrahim (muhammad.ibrahim@research.uwa.edu.au)

This work was supported by the Australian Research Council under Grant DP190102443.

ABSTRACT Accurate semantic segmentation of unstructured 3D point clouds requires large amount of annotated training data for deep learning. However, there is currently no free specialized software available that can efficiently annotate large 3D point clouds. We fill this gap by introducing PC-Annotate - a public annotation tool for 3D point cloud research. The proposed tool not only enables systematic annotation with a variety of fundamental volumetric shapes, but also provides useful functionalities of point cloud registration and the generation of volumetric samples that can be readily consumed by contemporary deep learning point cloud models. We also introduce a large outdoor public dataset for 3D semantic segmentation. The proposed dataset, PC-Urban is collected in a civic setup with Ouster LiDAR and labeled with PC-Annotate. It has over 4.3 billion points covering 66K frames and 25 annotated classes. Finally, we provide baseline semantic segmentation results on PC-Urban for popular recent techniques.

INDEX TERMS 3D point cloud, point cloud dataset, annotation tool, semantic segmentation, LiDAR.

I. INTRODUCTION

Point cloud semantic segmentation is a central problem in the real-world scene understanding. A point cloud encodes objects and scenes using their surface coordinates and enables accurate analysis based on their 3D shapes [1].Hence, point cloud semantic segmentation finds many important applications in emerging technologies such as self-driving vehicles, human-machine interaction, automatic surgery, and robot navigation. Deep learning has recently become the most widely adopted framework for segmenting 3D point clouds [2]–[7]. Deep neural networks have been shown to learn accurate computational models for this task. However, they can only do so with the availability of large amount of annotated training data.

The ground breaking performance of deep learning methods for point cloud semantic segmentation calls for easy access of 3D data annotation tools that can further this research direction by enabling efficient labeling of large training datasets. However, currently, there is no specialized annotation tool available that can be deployed on local machines for efficient annotation of large point clouds. Consequently, the research community generally resorts to commercial [8], [9] or online [10], [11] annotation tools, which are not only cost prohibitive, but also raise data privacy concerns.

The first major contribution of this article is the first-of-its-kind public 3D point cloud annotation tool which can be locally installed on users' machines to maintain data privacy. The tool can be installed on Windows, Linux and Mac operating systems.[1] The proposed tool, PC-Annotate provides a comprehensive labeling setup, while also offering complementary functionalities of point cloud registration and preparing data samples that can be directly consumed by state-of-the-art deep learning models. With a user-friendly graphics interface, PC-Annotate takes advantage of regular volumetric shapes (e.g. Cuboid, Cylinder, Sphere) and irregular free-hand geometric shapes to efficiently label large 3D point clouds. While annotating, it also provides easy to comprehend book-keeping and data loading functionalities. PC-Annotate also registers multiple LiDAR frames for simultaneous labeling of objects that appear in the multiple frames. The tool is specifically aimed to accelerate 3D point cloud annotation and research.

The second major contribution of the paper is a large outdoor annotated dataset for point cloud semantic

The associate editor coordinating the review of this manuscript and approving it for publication was Szidónia Lefkovits.

[1]Operating System versions released after 2016 are supported.

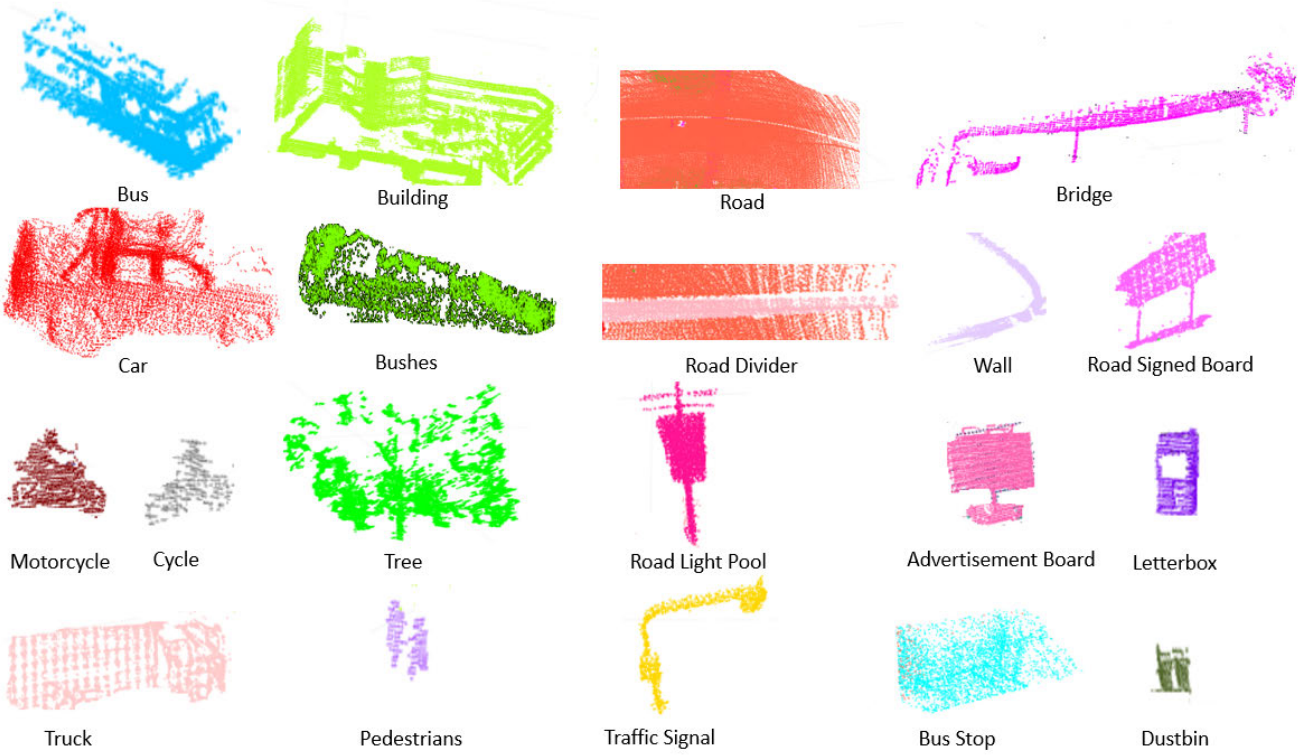


FIGURE 1. Representative examples of labeled samples for 20 (out of 25) classes from the PC-Urban dataset, annotated with PC-Annotate. Objects are selected from different scenes for illustration.

segmentation in urban environment. The proposed dataset, termed PC-Urban (Urban Point Cloud), is captured with an Ouster LiDAR sensor with 64 channels. The sensor is installed on an SUV that drives through the down town of Perth, Western Australia (WA), Australia. The dataset comprises over 4.3 billion points captured for 66K sensor frames. The labelled data is organised as registered and raw point cloud frames, where the former has different number of registered consecutive frames. We provide 25 class labels in the dataset covering 23 million points and ∼5K instances. Labeling is performed with PC-Annotate and can easily be extended by the end-users employing the same tool.[2] In Fig. 1, we show representative examples of annotated objects from different scenes for selected class labels. To demonstrate the compatibility of our tool with deep learning methods and establish a baseline for our dataset, we also report the performance of PointNet [2], PointNet++ [3] and PointConv [5] on our dataset. The results ascertain the existence of challenging practical scenarios in the proposed dataset.

The rest of the article is organized as follows. In Section II, we review the existing related literature. We discuss the proposed annotation tool in Section III. The proposed dataset is detailed in Section IV. We conduct experiments with our dataset in Section V. Finally, the article concludes in Section VI.

[2]The tool and dataset will be made public for crowd sourcing large-scale data annotation. Our lab will maintain and provide support for both.

II. LITERATURE REVIEW

We mainly survey the existing literature along the directions of annotation tools and datasets, and only briefly discuss popular deep learning methods for point cloud analysis.

A. ANNOTATION TOOLS

With the availability of LiDAR sensors, it is easy to collect large point clouds of outdoor scenes. However, labeling the data still remains a laborious and expensive task. It is a common practice in the point cloud research community to outsource data labeling to commercial companies such as Playment [8] and Scale IA [11]. The former provides labeling of both 2D and 3D data for computer vision related problems, charging researchers on per instance basis. Similarly, Scale IA [11] provides services of labeling various types of data including images, 3D point clouds, videos, audio, documents and text. It charges customers (i.e. researchers) based on the data size and also provides self-service to customers for labeling the data with their online tools.

BasicAI [9] is another example of a commercial annotation platform that provides data labeling services to the research community of Self-driving vehicles, Advanced Driver Assistance Systems (ADAS) and Mobile Robotics. Its annotation capabilities include 2D and 3D bounding boxes, image

segmentation, line marking, and sensor fusion. Employing these commercial solutions is generally financially infeasible for most research groups. Additionally, the labeling staff for these companies may not be well-trained for the precise labeling required for specific end-user problems. Another major issue comes in the form of security and privacy of the data, which is not guaranteed to be strictly preserved.

Another web-based online 3D point cloud labeling tool is provided by Supervisely [10]. The tool is free for research use. However, data security and privacy is still a major concern for such tools. For instance, [10] requires uploading the data on the server where it is annotated by the end-user, with no guarantee that the annotated data has not been copied. Moreover, the labeling capability of this tool is limited, keeping aside the requirement of uninterrupted Internet connection for successful labeling. Other indirect 3D point cloud labeling practices [12], [13] are too complicated for crowd sourcing. Another annotation tool has recently appeared in [14]. However, it provides limited functionalities for annotation. Currently, there is a clear need for a versatile and efficient 3D point cloud annotation tool that can be hosted locally by personal/lab machines to enable full data privacy. In this article, we address this pressing issue by introducing the required tool, PC-Annotate.

B. POINT CLOUD DATASETS

1) OUTDOOR DATASETS

Autonomous Driving Dataset (A2D2) [15] is a recent dataset captured using multiple LiDAR sensors. It comprises more than 40K labeled frames for semantic segmentation of point clouds in 38 categories. Among these frame, annotations of 3D bounding boxes for 12K frames are provided. Paris-Lille-3D [16] is another point cloud dataset developed by researchers at Mines ParisTech. It is captured in two cities of France, 'Paris' and 'Lille' with the Velodyne HDL-32E sensor. Semantic3D.Net [17] is created by researchers at ETH Zurich, Switzerland. It consists of ∼4 billion points collected from 30 scenes using terrestrial laser scanners.

The KTTI dataset [18] is developed to detect scene flow by building on the raw data of the existing KTTI dataset [19]. It captures 200 scenes each for training and testing, while providing labels for 8 object classes. Oxford RobotCar dataset [20] is acquired with 2D cameras and LiDAR sensors. It captures various scenes in the vicinity of Oxford, UK that include objects like roads, people, trees and buildings. For this data, ∼20 million images are collected in all weather conditions and 3D point cloud is generated from the images using a MATLAB tool. RueMonge2014 [21] is another popular outdoor dataset, which comprises scenes of ∼700 meters long street. In total, it provides 7 class labels for 0.9 million points. SemanticKITTI [14] is the latest version of the KITTI dataset for point cloud segmentation. Other recent outdoor datasets include Toronto-3D MLS [22] and TUM-MLS [23]. We also summarize the main properties of the above discussed outdoor datasets along other existing datasets in Table 3 of Section IV, where we also give detail of the proposed dataset for comparison.

2) INDOOR DATASETS

ScanObjectNN [24] is a recent example of indoor dataset that provides 15 class labels for ∼15K object instances. It comprises ∼700 unique scenes containing real-world objects. Similarly, ScanNet [25] provides 1,513 scans of various indoor environments, defining 20 object categories for semantic segmentation. The Stanford large-scale 3D Indoor Spaces (S3DIS) [26] captures five indoor units of three buildings that include office environment, exhibition center, conference rooms, restrooms, lobbies and hallways etc. There are 12 structural semantic objects for the recognition task. Stanford 2D-3D-Semantics [27] is another example of indoor dataset and provides labels for 13 object categories. Table 3 also summarizes the discussed indoor datasets.

3) SHAPE DATASETS

Instead of indoor/outdoor environments, shape datasets focus on individual objects. For instance, the FlyingThings3D dataset [28] consists of 25K stereo frames of everyday objects. It provides images of 200 static objects, and 32,872 object models for training and 3,055 models for testing. PartNet [29] contains 573K part instances with 26K 3D models of 24 objects. It is one of the largest dataset for full shape parts, providing dense point clouds of 26.6 billion points. ModelNet40 [30] is another popular example of the shape datasets, synthetically generated using a 3D CAD tool. It provides 40 object labels, in both aligned and non-aligned format.

C. METHODS

PointNet [2] is the pioneering deep learning model that directly consumes 3D point clouds. This method was extended to PointNet++ [3] that has the ability to account for global point features in addition to the local ones considered in PointNet. Another prominent recent method is PointConv [5], which allows efficient point cloud processing by capitalising on the prowess of convolution operation. In this work, we utilize the above mentioned approaches as representative techniques to establish a baseline for the proposed dataset. A large number of deep learning methods are emerging in the recent point cloud literature [1], [31]–[33], [4], [34]–[36], [7], [37]–[39], [30]. We refer interested readers to [40] for a comprehensive review.

III. PROPOSED ANNOTATION TOOL

To fill-in the gap of public 3D annotation tool for the research community, we introduce PC-Annotate - an efficient tool for labeling large point clouds. From the annotation of point clouds to the registration of raw frames and data preparation for deep models, PC-Annotate provides a variety of functionalities to accelerate point cloud research. It provides a user-friendly Graphical User Interface (GUI), see Fig. 2, which allows annotation with regular volumetric shapes

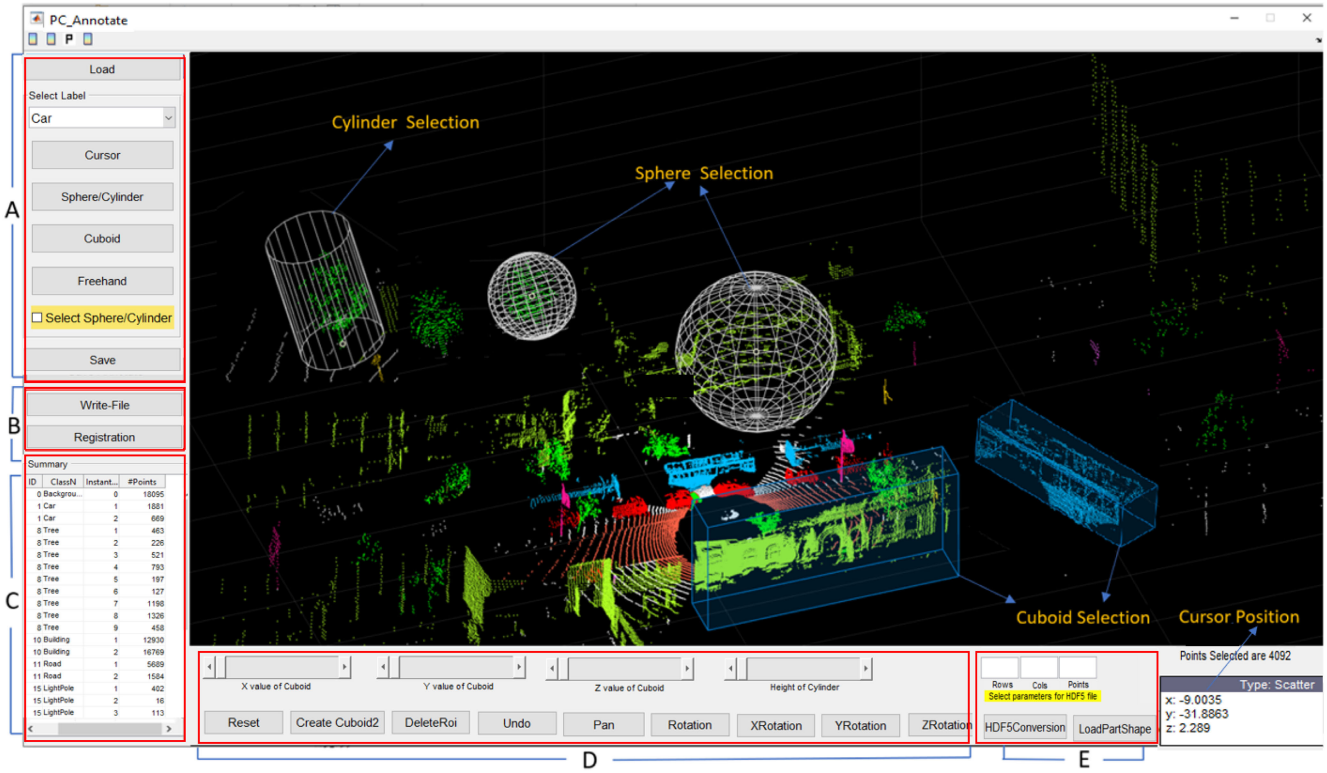


FIGURE 2. GUI of the proposed PC-Annotate: (A) Annotation ribbon, (B) Write and register ribbon, (C) Annotation summary panel, (D) Shape adjustment toolbar and viewing window control, (E) Processing ribbon. Refer to the text for explanation.

(Cuboid, Cylinder and Sphere) as well as irregular Freehand point selection. The shape sizes are easily controllable with sliding bars and mouse scroll. Regular geometric shapes allow efficient labeling by encapsulating objects, whereas, the Freehand selection enables more intricate annotation that suites isolated points. It selects points by hovering the mouse cursor with depressed mouse button. This results in defining a region of interest (RoI), within which all points can be simultaneously labeled. A large point cloud can easily be handled by this tool. For instance, we label a registered point cloud of the proposed dataset with 1.3 million points using the PC-Annotate. Below, we provide discussion on the tool based on the GUI divisions shown in Fig. 2.

A. ANNOTATION RIBBON (A)

This section of the GUI provides the core annotation functionalities, i.e. data loading, label selection, and saving the labels. After launching the tool, a raw/registered point cloud with .ply format or 3D points with their label text file (if previously labeled) can be loaded with the help of the Load button. Unlabeled loaded points appear white in the viewing window, which get their unique colors upon labeling. The class label selection can be made with a popup-menu that provides 32 common labels of interest for outdoor settings. For annotation, different geometric shapes can be accessed with the mouse scroll. Pressing the shape buttons in (A) allow storing the labels of the points circumscribed by the respective shapes. Between Sphere and Cylinder, the shapes can be toggled by clicking the Sphere/Cylinder check-box. We give further details on all buttons, and their associated short-keys in the supplementary material of the paper as a brief user-guide. Using the provided regular and irregular shapes, each instance of an object can be efficiently labeled, after which the Save button is pressed to store the instance information. A useful functionality of the tool is that incorrectly labeled points can be relabeled correctly since the last labeling over-rides the previous one.

B. WRITE AND REGISTER RIBBON (B)

This ribbon provides buttons for writing and registering point clouds. Once annotation of a point cloud is complete, PC-Annotate can write the labels to a hard drive by pressing the Write-File button or the shortcut key 'w'. This results in writing two text files, a *label file* and a *summary file*. The label file contains information on class IDs and class instances for each and every point. Class IDs are 0-indexed integer values, where 0 is reserved for the background. The summary file records the statistics of the labeled points, while noting the class IDs, instance IDs and number of annotated points for each instance. The writing operation can be performed at any time during the annotation process. The proposed tool is also capable of loading an already labeled or unfinished labeled point cloud by loading its corresponding saved label file.

This is a useful functionality to split the labeling process between different sessions or to simply verify saved labels.

Registration of raw point clouds is also enabled by this ribbon. Point cloud registration is a widely used operation that can help in simultaneous annotation of multiple point cloud frames. However, this functionality is normally ignored in existing labeling tools. In PC-Annotate, multiple point clouds can be registered prior to annotation. In many cases, this significantly improves the annotation efficiency. The proposed tool employs the popular Iterative Closest Point (ICP) algorithm [41] for the registration process. The precision of this process is higher than that of the high-end state-of-the-art LiDAR sensors.

C. ANNOTATION SUMMARY PANEL (C)

Appearing at the bottom left corner of Fig. 2, this panel displays the progress summary of the scene labeling. The summary is updated every time the Write-File button is pressed and provides information on Class name, Class ID, Instance ID and the number of points for that instance. It sorts the values in ascending order of the Class IDs.

D. SHAPE ADJUSTMENT TOOLBAR & VIEWING WINDOW CONTROL (D)

PC-Annotate provides regular geometric shapes such as Sphere, Cuboid, and Cylinder to circumscribe points for annotation. The Shape Adjustment Toolbar provides handles to adjust these shapes. We also provide shortcut keys for the shape adjustments (details in supplementary material). We provide two types of Cuboids. For the first, the cuboid is generated at the cursor position of a selected object by moving and of the three sliders in this toolbar. The sliders also govern the cuboid dimensions independent of each other. For the second, the cuboid is generated by pressing the 'Create Cuboid2' button for a default cuboid size ($5 \times 5 \times 5$). The mouse can then be used to adjust the dimensions by hovering the cursor over a plane of the cuboid and keeping the left mouse button pressed and dragging the cursor. For Sphere and Cylinder, their radius is controlled with the mouse wheel. The height of a Cylinder is controlled with the provided slide-bar.

It is possible to delete all the active geometric annotation shapes in the viewing window by pressing the 'DeleteRoi' button. Similarly, the 'Undo' button allows un-doing the last annotated instance. Easy control over the viewing window is a key attribute of effective annotation tools. Our tool provides this control with multiple buttons in the toolbar. For instance, panning of the point cloud can be activated by pressing the Pan button. Once panning mode is active, a scene can be moved by pressing and dragging the left mouse button upward or downward. Zoom in/out is also enabled with mouse control by double clicking the right/left buttons. The 'X,Y,ZRotation' buttons enable rotation along the respective axes, whereas the 'Rotation' button allows free rotation along all the axes with mouse movement, with left mouse button depressed. Finally, the 'Reset' button allows restoring the view and toolbar sliders to their default values.

E. PROCESSING RIBBON (E)

It is a common practice in deep learning literature to process point clouds as smaller volumetric samples extracted from large point clouds. The most common approach is to define an N×N area on the ground and slice a volume of point cloud over that area. Within that volume, a fixed number of points are selected for processing the sample with a deep learning model. Selecting 10×10 area and choosing 4,096 points per-sample, is among the most commonly occurring setups in the literature. In the current version of our PC-Annotate, the HDF5Converter button directly performs sampling of the point cloud in mutually exclusive blocks of base area 10×10, such that each sample has 4,096 points. The HDF5 files resulting from this button can be readily used as input to popular networks like PointNet and PointNet++. In Fig. 3, we show representative samples resulting from PC-Annotate. In these samples, if a volume contains more than 4,096 points, the required 4,096 points are randomly sampled. On the other hand, volumes with fewer points repeat random points to achieve the desired number. This strategy is inspired by the literature [2]. The number of samples, points per-sample and values per-point can also be adjusted in the tool by the users. The current version of the proposed annotation tool can also load part-shape labels and their corresponding points. It gives them colors according to their labels, allowing for `.seg` file loading. We summarise the process flow of the proposed PC-Annotate in Fig. 4 for the higher level operational understanding of the tool.

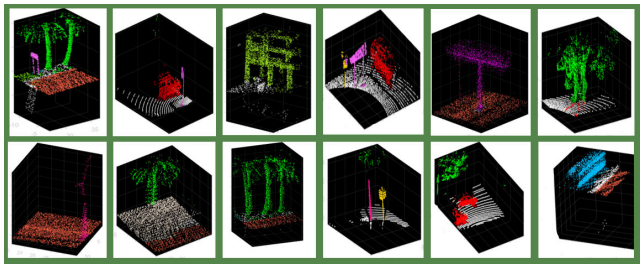


FIGURE 3. Representative 10 × 10 samples of 4096 points each generated by PC-Annotate for the PC-Urban dataset.

F. PC-ANNOTATE ADVANTAGES OVER CURRENT PRACTICES

Apart from commercial tools, there are no specialized annotation tools for 3D point cloud. Hence, the research community resorts to adopting complicated practices to resolve this issue. For instance, DublinCity dataset [12] is annotated with CloudCompare, which is actually designed for point cloud visualization only, not annotation. Its main functionality is to divide a point cloud into segments, resulting in cumbersome annotation of geometrically complex shapes. Unlike PC-Annotate, CloudCompare also does not have proper book keeping of labels and annotation statistic. Another labeling tool has been introduced in [14]. However, it ignores multiple important annotation shapes such as cylinder and freehand.

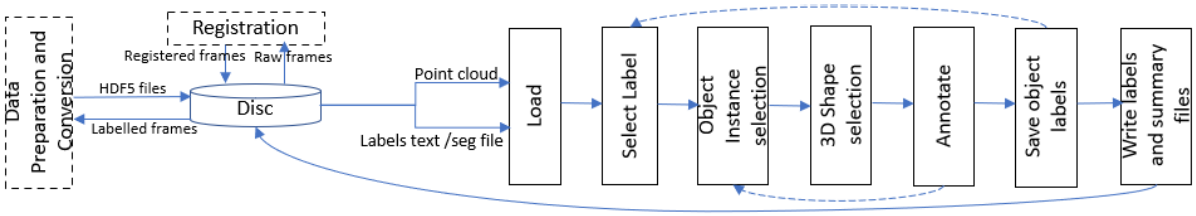


FIGURE 4. Process flow of PC-Annotate: The tool allows reading raw frames from the computer disc and can write registered frames, HDF5 files and (complete/incomplete) labeled frames. After loading the data, the desired instance label is selected. An object is (possibly iteratively) annotated with regular or irregular shapes (or their combinations). After each instance annotation, the label information is stored. The process is repeated for as many objects as desired. The final label and summary file is then written to the disc.

TABLE 1. Annotation time (in minutes) of four users to annotate seven random scenes (S-1 to S-7) of the proposed outdoor PC-Urban dataset using the PC-Annotate tool. All the users were new to the tool with no prior experience for the task.

Users	S-1	S-2	S-3	S-4	S-5	S-6	S-7	Avg
User 1	20.3	18.0	17.2	16.4	14.0	15.3	15.0	16.6
User 2	17.0	16.0	15.3	14.3	13.1	14.0	12.0	14.5
User 3	21.0	20.4	18.3	18.0	17.0	16.3	16.0	18.1
User 4	19.0	18.0	16.0	16.0	15.0	15.0	15.3	16.3
All	Average Annotation time							**16.4**

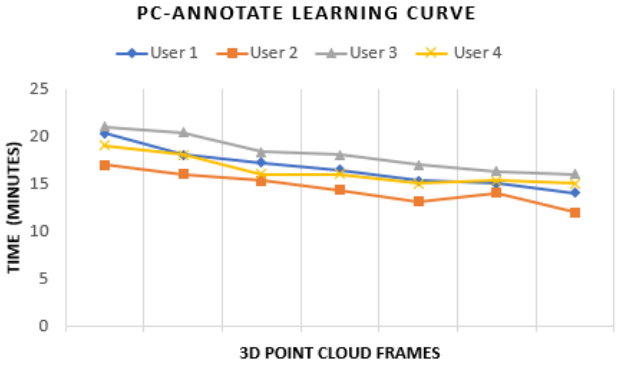


FIGURE 5. PC-Annotate learning curve for four new users. Each user annotates seven different frames with similar looking scenes.

Also, it does not offer point cloud registration functionality. It also fails to provide annotation statistics during the process. These facts makes it hard for an untrained user to adapt to the tool. Another annotation method is used in [13] where 3D labels are obtained with the help of 2D RGB images. This approach is computationally expensive and inefficient. For instance, the average annotation time reported by [13] for an indoor scene is 30 minutes. Not to mention, acquiring 3D labels from 2D data also compromises the accuracy of 3D annotation. Xie *et al.* [42] also devised an opposite strategy of annotating images with by annotating 3D reconstruction static scene elements using rough bounding primitives.

The proposed PC-Annotate is a specialized, efficient and user friendly annotation tool for 3D point cloud semantic segmentation. It allows accurate annotation of complex geometric shapes at all scales. PC-Annotate uses hardware accelerated OpenGL graphics pipeline for computational efficiency. To demonstrate user friendliness and efficiency of our tool, we report annotation times taken by four random users (undergrad students) on seven random scenes of PC-Urban (proposed outdoor dataset) in Table 1. The average annotation per scene of PC-Urban is found to be 16.4 minutes for those scenes. The learning curve of PC-Annotate for the users is shown in Fig 5. The graphs show that PC-Annotate is user-friendly. The annotation time for the new users gets reduced after labeling few frames. Table 2 reports the annotation time with the available geometric shapes and operations of PC-Annotate for different objects. These results conclude that on average, over 3K points can be labeled by PC-Annotate in about well under half a minute with the available functionalities, which is highly desirable.

IV. PROPOSED PC-URBAN DATASET

As the second major contribution, we introduce a large scale real-world outdoor point cloud dataset. The proposed dataset is partially annotated and released with the annotation tool for crowd source labeling.

The data is collected within 10km of central Perth city in Western Australia. The collection is made under a variety of day times and conditions by driving a LiDAR mounted SUV through various routes, see Fig. 6. The collected data size is approximately 50 GB. The data is organized into unlabelled and labelled 3D point clouds as shown in Fig 7. The unlabelled data is provided in.PCAP file format, which is the direct output format of the used Ouster LiDAR sensor. Raw frames are extracted from the recorded.PCAP files in the form of Ply and Excel files using the Ouster Studio Software. Labelled 3D point cloud data consists of registered or raw point clouds. A labelled point cloud is a combination of Ply, Excel, Labels and Summary files. A point cloud in Ply file contains x,y,z values along with color information. An Excel file contains x,y,z values, Intensity, Reflectivity, Ring, Noise, and Range of each point. These attributes can be useful in semantic segmentation using deep learning algorithms. The Label and Label Summary files have been explained in the previous section. Our one GB raw data contains nearly 1,300 raw frames, whereas 66,425 frames are provided in the dataset, each comprising 65,536 points. Hence, 4.3 billion points captured with the Ouster LiDAR

TABLE 2. Annotation time for different objects using sphere, cylinder, cuboid and freehand functionalities provided by PC-Annotate. The values are reported in seconds.

	Tree	Building	Road	Traffic Signal	Signboard	Light Pole	Car	Bus	Avg
# of Points	819	5261	15124	724	340	228	1532	2179	3,275.92
Sphere	20.29	27.56	48.70	15.31	05.96	12.02	13	10.23	19.13
Cylinder	26.48	40.57	74.53	11.64	07.65	11.75	24.85	16.02	26.68
Cuboid	27.19	38.76	41.54	24.48	18.70	19.96	14.42	20.92	25.74
Freehand	25.33	48.89	45.42	12.04	11.98	08.33	11.75	13.97	22.21
Avg	24.82	38.94	52.54	15.86	11.07	13.01	16.00	15.28	**23.44**

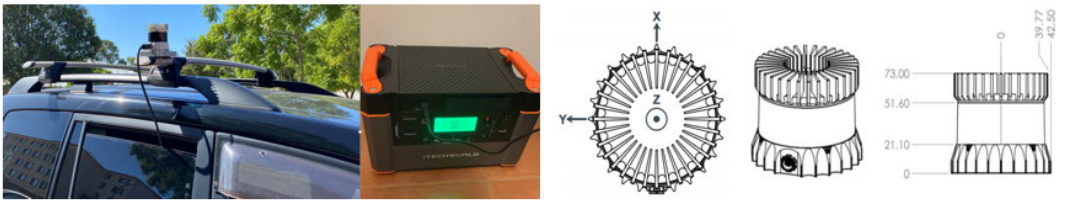


FIGURE 6. From left to right. Ouster LiDAR installed on the roof of SUV. Portable power source. Sensor coordinate frame and dimensions (mm).

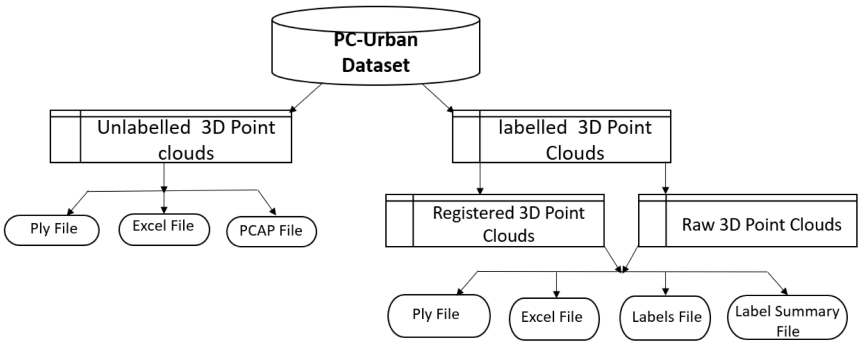


FIGURE 7. Data organization in PC-Urban dataset containing both unlabelled and annotated 3D point clouds.

sensor are provided. Annotation of 25 general outdoor classes are provided, which include car, building, bridge, tree, road, letterbox, traffic signal, light-pole, rubbish bin, cycles, motorcycle, truck, bus, bushes, road sign board, advertising board, road divider, road lane, pedestrians, side-path, wall, bus stop, water, zebra-crossing and background. With the released data, a total of 106 scenes are annotated which include both raw and registered frames, explained in more detail below.

A. LiDAR SETUP FOR DATA COLLECTION

The data is captured using a 64 channel Ouster LiDAR mounted on an SUV and powered from a portable 220V AC source through an interface box accompanied with the sensor. The sensor is fitted on the right hand roof rack (driver side) about 1.5m above the ground and 12cm above the vehicle roof (see Fig. 6) and connected via Gigabit Ethernet cable to a laptop (2.2GHz i7 processor, 16GB RAM) using Ouster studio software running over Ubuntu 18.04 operating system.

1) SENSOR SPECIFICATIONS

To fully comprehend the introduced dataset, it is vital to know the specifications of the employed sensor. The Ouster LiDAR used in this work has 64 channels, i.e. vertical resolution is 64, while its horizontal resolution is configurable to 512, 1,024 or 2,048. We use 1,024 as the horizontal resolution in our dataset. The vertical Field of View (FoV) of the sensor is 45°: (+22.5° to −22.5°), while its horizontal FoV is 360°. Rotation rate of the sensor is configurable to 10 or 20Hz. We use 10Hz, which provides frames at 10 fps. The wavelength of the sensor laser is 865nm. It collects 1,310,720 points per second and per point sensed data includes range values, intensity, reflectivity, ambient, channel, azimuth angle, and timestamp. This information is available in the provided raw data. The power consumption of the sensor is 14 - 20 W and its operating voltage is 22 - 26V DC. A power box converts 220V AC to the required DC range. The sensor data connection is via a Gigabit Ethernet cable employing the UDP protocol.

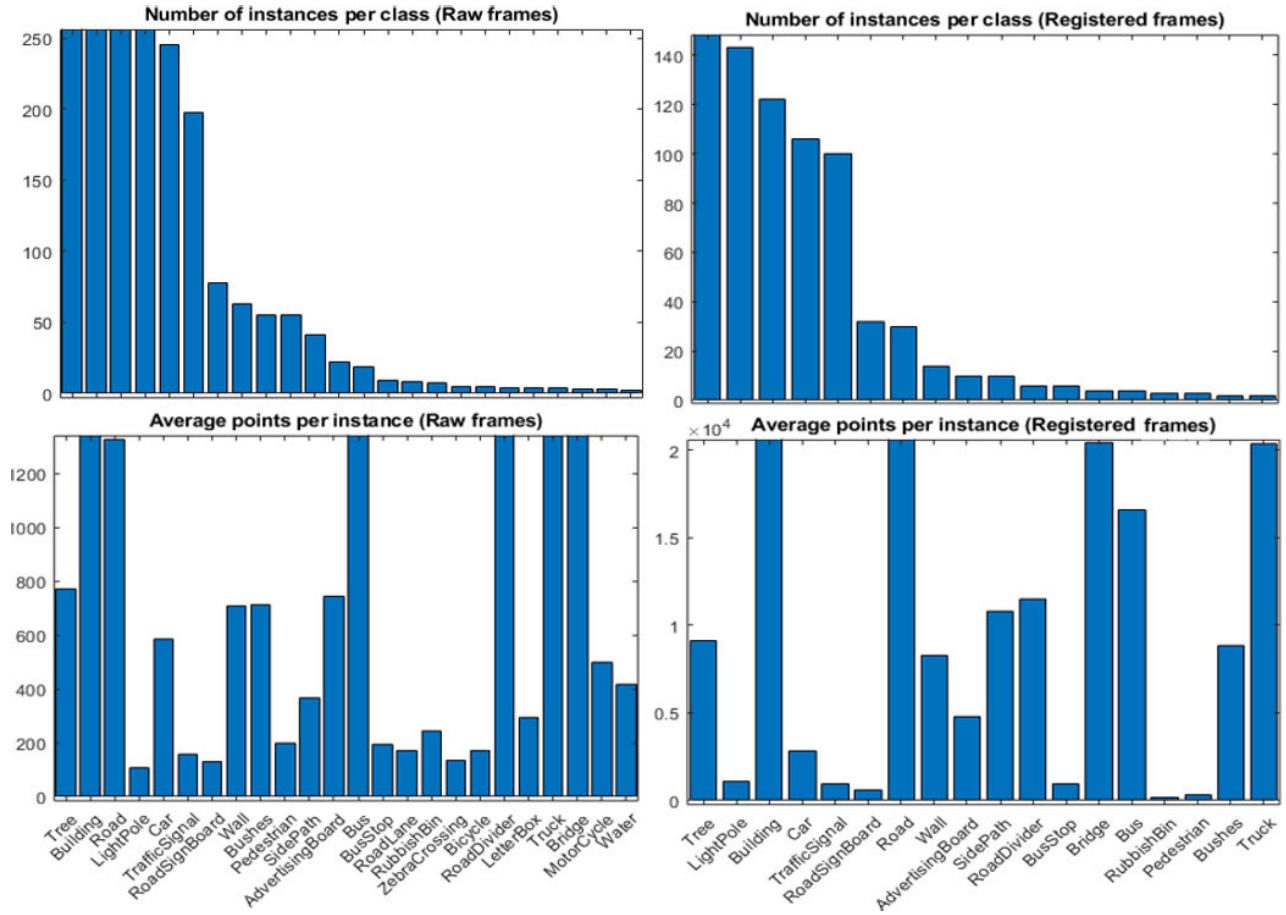


FIGURE 8. Top: (left) Number of instances per class for raw frames. For clarity, maximum limit of 250 instances is used in the graph. The actual number of instances for tree, building, road, and light-pool are 1044, 599, 352 and 305, respectively. Bottom: (left) Average number of points per instance per class for raw frames. A limit of 1300 points is used. Average points per instance for bridge, road divider, truck, building and bus are 2871, 2765, 2173 1893 and 1824, respectively. Top: (right) Number of instances per class for registered frames. The Tree class has 605 instances. Bottom: (right) Average number of points per instance per class for registered frames. For building, road, bridge, truck and bus, the average points per instance are 40.7K, 40K, 20.4K, 20.3K and 16.5K, respectively. For the registered point clouds, 18 class labels are provided, whereas 24 labels are provided for the raw frames.

2) X,Y,Z-COORDINATES CALCULATION FROM RANGE DATA

A 3D point is generally represented as its X,Y,Z-coordinate values. The LiDAR provides range, encoder count, beam altitude angle, and beam azimuth angle. The corresponding X,Y,Z values for a point are computed as:

$$X = r\sin(2\pi(\frac{e_c}{90112} + \frac{\theta[i]}{360}))\cos(2\pi\frac{\phi[i]}{360}), \quad (1)$$

$$Y = -r\sin(2\pi(\frac{e_c}{90112} + \frac{\theta[i]}{360}))\cos(2\pi\frac{\phi[i]}{360}), \quad (2)$$

$$Z = r\sin(2\pi\frac{\phi[i]}{360}), \quad (3)$$

where r is the range in milli-meter, e_c is the encoder count, $\theta[i]$ is the beam azimuth angle of the i^{th} channel, and $\phi[i]$ is the beam altitude angle of the i^{th} channel. The computed X,Y,Z values for all points are provided in the proposed PC-Urban dataset for direct usage as point clouds.

B. RAW ANNOTATED FRAMES

The proposed PC-Urban dataset contains annotations for both raw and registered frames. In the provided data, 83 raw point cloud frames are annotated using the proposed annotation tool. This resulted in 3,131 instances with approximately 6 million points for 25 classes. In Fig. 8, the top left graph shows the number of instances per class for raw frames while the bottom right graph shows the average number of points per instance per class for the raw labeled frames. The data is collected in an urban environment where large number of trees, light-poles, buildings, cars and traffic signals are found, resulting in more instances for these classes. There is a single instance per frame for the background class (not shown) which has approximately 30k points. Smaller objects such as cycle, light-pole, letterbox, rubbish bin, pedestrians, road-sign boards, and traffic signals generally have around 200 average points per instance. In Fig. 9, the first three columns show representative examples of labeled raw frames of PC-Urban dataset annotated with PC-Annotate. We also

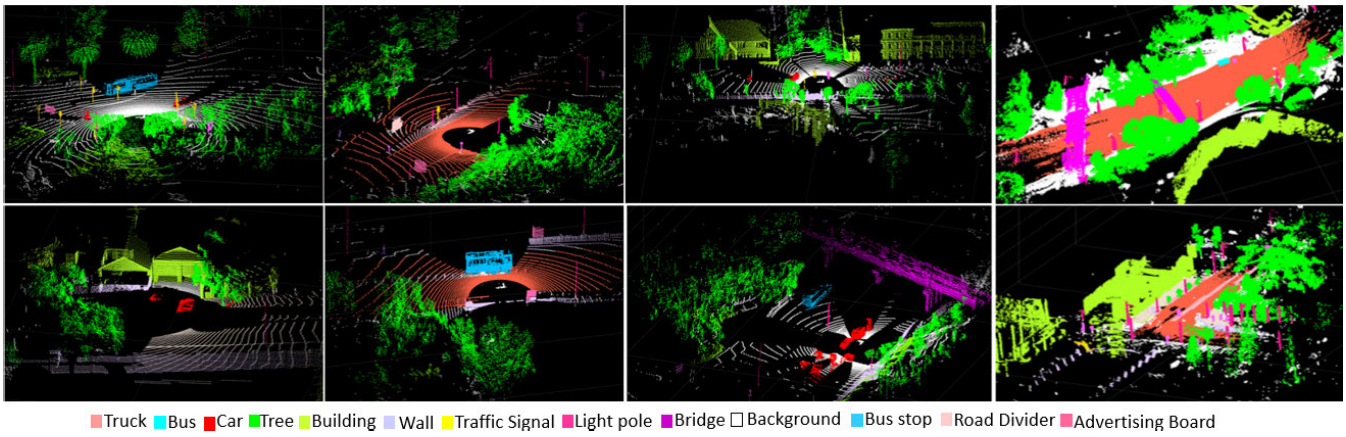


FIGURE 9. Representative labeled frames of PC-Urban dataset, labeled with PC-Annotate. The last column shows registered labeled frames.

TABLE 3. Popular contemporary 3D point cloud datasets for semantic segmentation task. Indoor datasets are highlighted red. Due to their remote relevance, we do not include CAD based synthetic shape datasets, e.g., Flying3D [28], PartNet datasets [29] and ModelNet40 [30].

Dataset	Classes	Points	Sensor	Type
A2D2 [15]	38	unknown	LiDAR sensor	Outdoor
Paris-Lille-3D [16], IJRR18	50	143.1M	Velodyne HDL-32E	Outdoor
Toronto-3D MLS [22], CVPRW20	8	78.3M	32-line LiDAR	Outdoor
TUM-MLS [23], Remote Sens.20	8	1.7 B	Dual Velodyne HDL-64E	Outdoor
Semantic3D.Net [17], arXiv17	8	4B	Terrestrial Laser	Outdoor
KITTI [18], CVPR15	8	-	Velodyne 3D and laser scanner	Outdoor
Oxford RobotCar [20], IJRR16	-	-	LiDAR and Cameras	Outdoor
RueMonge2014 [21], ECCV14	7	0.4M	sfM Reconstruction	Outdoor
TUM City Campus [44], ISPRS17	9	1.7B	Velodyne HDL-64E	Outdoor
iQmulus [45], Comp. & Graphics	**50**	300M	Riegl LMS-Q120i	Outdoor
Paris-rue-Madame [46], arXiv14	17	20M	Velodyne HDL-32 lidar	Outdoor
Oakland [47], CVPR09	11	1.6M	SICK LMS lidar	Outdoor
SemanticKITTI [14], ICCV19	28	-	Velodyne HDL-64E	Outdoor
ScanObjectNN [24], ICCV19	15	15.3M	Structure Sensor, CAD model	Indoor
ScanNet [25], CVPR17	20	-	Structure Sensor	Indoor
S3DIS [26], CVPR16	12	-	Matterport	Indoor
Stanford 2D-3D-Semantics [27], arXiv17	13	-	Matterport 3Dcamera	Indoor
PC-Urban dataset (Proposed)	22	**4.3B**	Ouster Lidar (64 Channels)	Outdoor

provide more raw labelled samples in supplementary material of the article.

C. REGISTERED LABELED FRAMES

We provide 23 annotated registered frames in our dataset, which are generated from ∼3,400 raw frames. These frames are registered and labeled with PC-Annotate, providing over 16 million annotated points for 1,225 instances. The density of a registered frame depends on the speed of the vehicle used for data collection. The most dense frame is produced by registering 210 raw frames, whereas the least dense one is generated by registering 50 consecutive raw frames. Registration process automatically removes the moving objects from a scene such as vehicle, bicycle, and pedestrians. The used popular ICP algorithm for registration has the ability to remove all moving objects from a 3D point cloud scene during the process. Thus, the number of class labels for registered frames is less than raw frames in our dataset. In Fig. 8, the top right graph shows the number of instances per class in the registered frames while the bottom right graph represents the average number of points per instance per class. Again, we do not show the background class, containing 127K points for the sole instance. Notice the variation of labels along the x-axis of the graphs for raw and registered frames. Along providing denser point clouds, the registered frames also alter the distribution of instances per class and average points per instance distribution, compared to the raw frames adding an extra dimension of variability to the registered frames. In Fig. 9, the last column shows an example of registered frames from PC-Urban where objects from multiple frames have been labeled simultaneously with PC-Annotate. More registered labelled samples are also provided in the supplementary material of the article. We also summarize the statistics of contemporary datasets in Table 3 to demonstrate the larger overall size of the proposed data.

D. PC-URBAN COMPARISON WITH OTHER EXISTING DATASETS

We compare the proposed PC-Urban dataset with the existing outdoor point cloud datasets in Table 3. Besides variety of

the urban scenes, our data provides improvements over the existing datasets in terms of number of points acquired and the used sensor. For instance, the LiDAR sensor used in nuSense [43] has a vertical FOV of 40° and beam size 32, which are lower than 45° FOV and 64 beam size of our sensor. Similarly, the vertical FOV of the LiDAR sensor used in semanticKITTI [14] is 26.9°. Moreover, the latest public datasets nuScenes [43] and A2D2 [15] are specifically collected for autonomous vehicle scenarios. Our data is more generic as it also contains general purpose classes such as bushes, bus stop, trees, letterbox, dustbin, and advertising board etc. In terms of size, only Semantic3D.Net [17] is comparable to our dataset. However, it provides only 8 class labels. DublinCity [12] and TUM City Campus [44] provide over 1 billion points, however with only 13 and 9 class labels, respectively. We provide labels of 25 classes, while offering 4.3 billion points. Other datasets such as iQmulus [45], RueMonge2014 [21] and Paris-rue-Madame [46] are generated using 3D reconstruction methods using 2D sensors or primitive LiDAR sensors. The used state-of-the-art LiDAR sensor, larger number of labels, and the large data size are highly desirable qualities of the proposed PC-Urban to advance research in 3D point cloud analysis using deep learning.

V. EXPERIMENTS

To provide a baseline, we evaluate the performance of three popular deep learning methods for point cloud segmentation on our data, i.e. PointNet [2], PointNet++ [3] and PointConv [5]. These experiments also provide a suggestive setup for PC-Annotate and our data. The results highlight the challenging nature of the proposed dataset.

A. SETTINGS

From the provided annotated data, we randomly select 127k samples of 4,096 points, generated with our annotation tool. Notice that we are exploiting PC-Annotate for model 'evaluation', which is an intended functionality of the proposed tool. We use 96K samples (i.e. 75%) for training and 31K samples (i.e. 25%) for testing. The model training uses 5% of the training data for validation. The number of points for training is approximately 400 million, which was found sufficient for the used 3D point cloud networks to be trained well. We use public Tensorflow implementations for the used techniques. All the models are trained for 50 epochs, while following the original works for the remaining training setup.

B. EVALUATION METRICS

For the semantic segmentation of 3D point cloud, we follow the standard evaluation metrics of mean Intersection over Union (mIoU), Accuracy (Acc.) and Average Class Accuracy (Avg acc.). The definitions of the metrics are:

$$IoU_i = \frac{TP_i}{TP_i + FP_i + FN_i} \tag{4}$$

TABLE 4. Semantic segmentation results of popular techniques on the proposed and existing 3D point cloud datasets. We report the mean Intersection over Union (mIoU) and average class accuracy (Avg. acc.) in %, for comparison.

Point cloud Dataset	Method	mIoU	Avg. acc.
2D-3D-Semantics [27]	PointNet [2]	47.71	24.24
	PointNet++ [3]	53.2	-
ScanNet [25]	PointNet++ [3]	33.9 [5]	-
	PointConv [5]	55.60	-
SemanticKITTI [14]	PointNet [2]	14.6	-
	PointNet++ [3]	20.1	-
Semantic3D [17]	PointNet++ [3]	63.1 [40]	-
PC-Urban(proposed)	PointNet [2]	12.1	16.5
	PointNet++ [3]	17.3	26.3
	PointConv [5]	37.6	50.2

$$mIoU = \frac{1}{n}\sum_{i=1}^{n} IoU_i \tag{5}$$

$$Acc_i = \frac{TP_i}{TP_i + FP_i} \tag{6}$$

$$Avg\ acc. = \frac{1}{n}\sum_{i=1}^{n} Acc_i \tag{7}$$

where 'i' indicates the i^{th} class of 'n' total classes, TP, FP, FN respectively denote the true positive, false positive and false negative.

C. RESULTS

In Table 4, we summarise the results of our experiments for point cloud semantic segmentation. The table also includes results of the evaluated techniques for other existing outdoor datsets for comparison. We report performance for the commonly used evaluation metrics of mean Intersection over Union (mIoU) and average class accuracy (Avg. acc.) for segmentation. As can be seen, the values for these metrics for the proposed dataset are fairly low compared to the existing datasets. This is mainly due to the variety of objects and challenging conditions captured in real-life outdoor setup in the proposed data. The table reports results for the other datasets from the literature. The quantitative results affirm the presence of hard segmentation scenarios in the proposed dataset. In Table 5, we provide performance details for PointNet++ and PointConv for representative classes on our data. The performance is reported in terms of Accuracy / IoU (%), using two significant figures. It can be noticed that PointConv [5] achieves better results on our dataset since PointConv models spatial correlations between the points early in the network. This indicates appropriate preservation of spatial patterns in the data with the chosen sensor specifications and adopted setup. We can also notice that the accuracy of the techniques are particularly higher for 'Tree' and 'Building' classes. This owes to the larger number of points and instances for these classes in the training data.

D. WHOLE-SCENE PREDICTION VISUALISATION

For illustration, the predicted output of PointNet, PointNet++ and PointConv on a representative whole scene is shown

TABLE 5. Class-wise segmentation performance of PointNet++ and PointConv for representative classes of PC-Urban. The results are shown as Accuracy / IoU (in %).

	Car	Tree	Building	Bus	Light pole	Traffic signal	Pedestrian	Wall	Advert-Board
Num of instances	61	235	32	8	67	34	20	15	10
Avg points per instance	762	701	3.7K	2.5K	86	181	268	1.7K	360
PointNet++ [3]	48 / 30	46 / 29	34 / 21	2 / 1.2	0 / 0	61 / 11	0 / 0	0 / 0	0 / 0
PointConv [5]	91 / 70	76 / 69	93 / 66	63 / 59	55 / 45	70 / 49	08 / 06	53 / 43	20 / 03

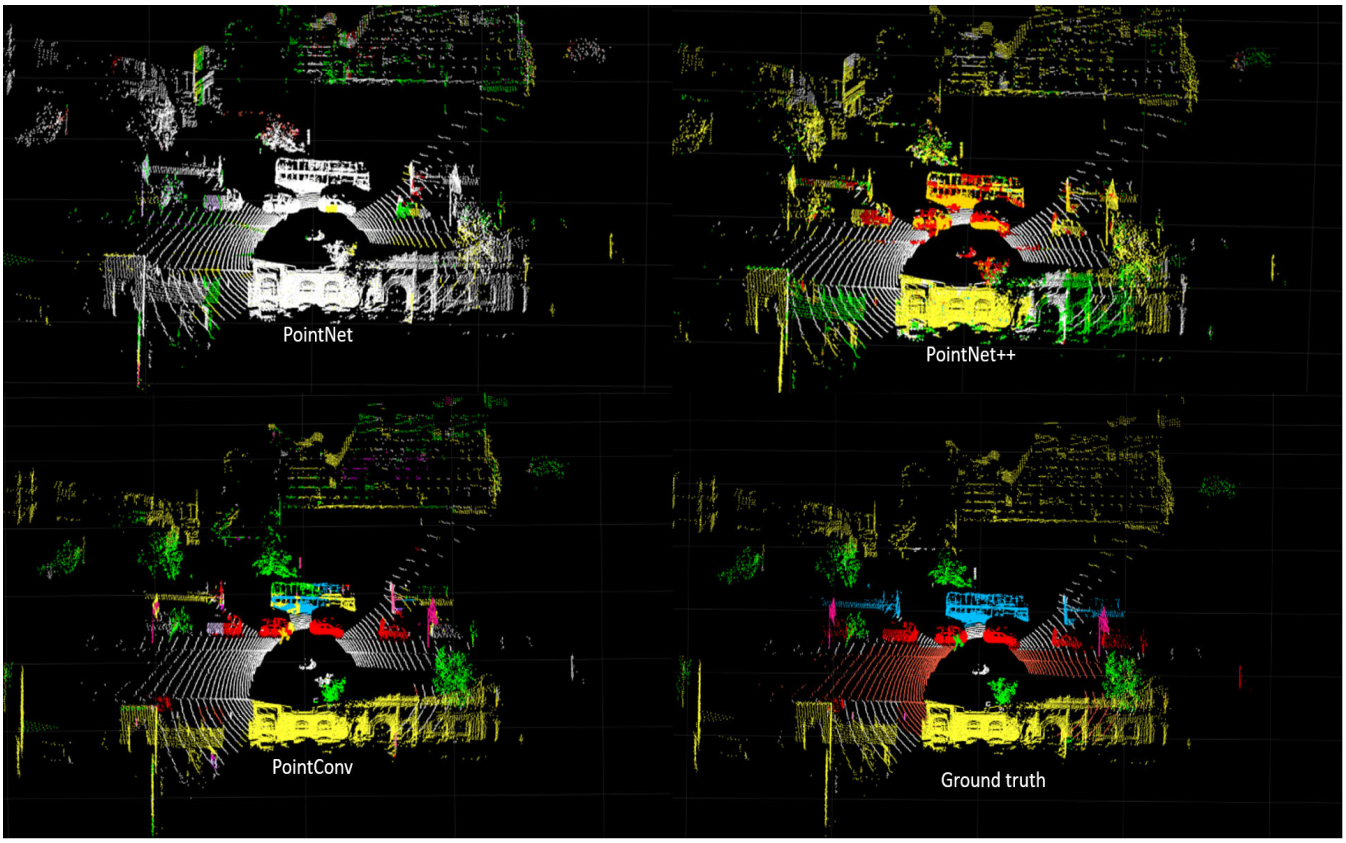


FIGURE 10. Representative examples of whole scene label predictions with the existing methods. Note: The colors of some classes have been altered for better visualization.

in Fig. 10. To predict, the scene is first divided into small ground areas (6×6), over which the volumetric slice is used for prediction. Unlike training and validation data where we split a scene by random selection of the floor area; here we sequentially divided the whole scene into mutually exclusive cubes and passed them through the networks for prediction. The predicted labels for the cubes were then integrated to generate the whole scene. Due to this minor change in the input data, the mean IoU and average class accuracy slightly dropped for the visualised scenes.

VI. CONCLUSION

Training effective point cloud deep learning models requires large amount of annotated data. However, currently, there is no effective public tool to annotate large point cloud datasets. This paper filled this gap by introducing PC-Annotate - a user friendly comprehensive public annotation tool for 3D point clouds. The proposed tool not only enables efficient labeling of large point clouds, but also provides functionalities of registering raw frames for simultaneous labeling and data preparation for deep learning models. We also introduced PC-Urban - a challenging real-world large scale point cloud dataset for urban environment. The dataset provides 4.3 billion points for 66K frames. Using PC-Annotate, we labeled 25 classes in the dataset. We also provided baseline results for point cloud semantic segmentation for our dataset using three popular deep learning techniques. The annotation tool and data are aimed at crowd sourcing the labeling of large-scale point cloud datasets. We plan to extend the labelled dataset further in the future.

REFERENCES

[1] M. A. Uy and G. H. Lee, "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4470–4479.

[2] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[3] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[4] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.

[5] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9621–9630.

[6] X. Liu, C. R. Qi, and L. J. Guibas, "FlowNet3D: Learning scene flow in 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 529–537.

[7] A. Komarichev, Z. Zhong, and J. Hua, "A-CNN: Annularly convolutional neural networks on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7421–7430.

[8] T. Mangukiya. (2020). *Data Labeling Platform for Computer Vision*. Accessed: Feb. 3, 2020. [Online]. Available: https://playment.io/3D-point-cloud/

[9] BasicIA. *Welcome to the Unified Platform for AI/ML Data Collection, Data Labeling, Outsourced Services, and Iterative Model Training*. Accessed: Feb. 3, 2020. [Online]. Available: https://www.basic.ai/

[10] Supervise.ly. (2019). *Releasing First Online 3D Point Cloud Labeling Tool in Supervisely*. [Online]. Available: https://medium.com/deep-systems/releasing-first-online-3d-point-cloud-labeling-tool-in-supervisely-4faca42b5d6e

[11] C. C. G. Brockman. (2020). *The Advanced Annotation Platform for 3D Sensor Data*. Accessed: Feb. 3, 2020. [Online]. Available: https://scale.com/3d-sensor-fusion

[12] S. M. I. Zolanvari, S. Ruano, A. Rana, A. Cummins, R. E. D. Silva, M. Rahbar, and A. Smolic, "DublinCity: Annotated LiDAR point cloud and its applications," 2019, *arXiv:1909.03613*. [Online]. Available: http://arxiv.org/abs/1909.03613

[13] D. T. Nguyen, B.-S. Hua, L.-F. Yu, and S.-K. Yeung, "A robust 3D-2D interactive tool for scene segmentation and annotation," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 12, pp. 3005–3018, Dec. 2018.

[14] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9297–9307.

[15] J. Geyer, Y. Kassahun, T. Fernandez, M. Mahmudi, M. Jänicke, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, and P. Schuberth. (2019). *A2D2: Aev Autonomous Driving Dataset*. [Online]. Available: http://www.a2d2.audi

[16] X. Roynard, J.-E. Deschaud, and F. Goulette, "Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification," *Int. J. Robot. Res.*, vol. 37, no. 6, pp. 545–557, May 2018.

[17] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "Semantic3D.Net: A new large-scale point cloud classification benchmark," 2017, *arXiv:1704.03847*. [Online]. Available: http://arxiv.org/abs/1704.03847

[18] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. (2015). *The Kitti Vision Benchmark Suite*. [Online]. Available: http://www.cvlibs.net/datasets/kitti

[19] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3354–3361.

[20] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford RobotCar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, Jan. 2017.

[21] H. Riemenschneider, A. Bódis-Szomorú, J. Weissenberg, and L. Van Gool, "Learning where to classify in multi-view semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2014, pp. 516–532.

[22] W. Tan, N. Qin, L. Ma, Y. Li, J. Du, G. Cai, K. Yang, and J. Li, "Toronto-3D: A large-scale mobile LiDAR dataset for semantic segmentation of urban roadways," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 202–203, doi: 10.1109/CVPRW50498.2020.00109.

[23] J. Zhu, J. Gehrung, R. Huang, B. Borgmann, Z. Sun, L. Hoegner, M. Hebel, Y. Xu, and U. Stilla, "TUM-MLS-2016: An annotated mobile LiDAR dataset of the TUM city campus for semantic point cloud interpretation in urban areas," *Remote Sens.*, vol. 12, no. 11, p. 1875, Jun. 2020.

[24] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1588–1597.

[25] M. Niessner, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and A. Dai, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5828–5839.

[26] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1534–1543.

[27] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2D-3D-Semantic data for indoor scene understanding," 2017, *arXiv:1702.01105*. [Online]. Available: http://arxiv.org/abs/1702.01105

[28] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4040–4048.

[29] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 909–918.

[30] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[31] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional ShapeContextNet for point cloud recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4606–4615.

[32] H. Deng, T. Birdal, and S. Ilic, "PPFNet: Global context aware local features for robust 3D point matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 195–205.

[33] J. Li, B. M. Chen, and G. H. Lee, "SO-Net: Self-organizing network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 9397–9406.

[34] Y. Zhao, T. Birdal, H. Deng, and F. Tombari, "3D point capsule networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1009–1018.

[35] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12697–12705.

[36] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7652–7660.

[37] H. Lei, N. Akhtar, and A. Mian, "Octree guided CNN with spherical kernels for 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9631–9640.

[38] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4558–4567.

[39] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining point cloud local structures by kernel correlation and graph pooling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4548–4557.

[40] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jun. 29, 2020, doi: 10.1109/TPAMI.2020.3005434.

[41] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *Proc. SPIE*, vol. 1611, pp. 586–606, Apr. 1992.

[42] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger, "Semantic instance annotation of street scenes by 3D to 2D label transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3688–3697.

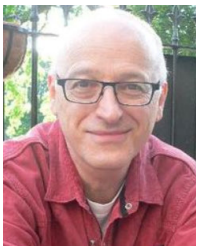
[43] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11621–11631.

[44] J. Gehrung, M. Hebel, M. Arens, and U. Stilla, "An approach to extract moving objects from MLS data using a volumetric background representation," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 4, p. 107, May 2017.

[45] B. Vallet, M. Brédif, A. Serna, B. Marcotegui, and N. Paparoditis, "TerraMobilita/iQmulus urban point cloud analysis benchmark," *Comput. Graph.*, vol. 49, pp. 126–133, Jun. 2015.
[46] A. Serna, B. Marcotegui, F. Goulette, and J.-E. Deschaud. (2014). *Paris-Rue-Madame Database: A 3D Mobile Laser Scanner Dataset for Benchmarking Urban Detection, Segmentation and Classification Methods*. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00963812/document
[47] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin Markov networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 975–982.

MUHAMMAD IBRAHIM received the B.S. degree in computer systems engineering from the University of Engineering and Technology (UET), Peshawar, Pakistan, and the M.Sc. degree in mobile and satellite communication from the University of Bradford, U.K. He is currently pursuing the Ph.D. degree in computer science with The University of Western Australia. Previously, he served as an Assistant Professor with the Department of Electrical Engineering, National University of Computer and Emerging Sciences (NUCES), Peshawar. His research interests include semantic segmentation of 3D point cloud, data sciences, machine learning, object detection, localization and tracking, and action recognition.

NAVEED AKHTAR received the master's degree in computer science from Hochschule Bonn-Rhein-Sieg, Germany, and the Ph.D. degree in computer vision from The University of Western Australia (UWA). He is currently an Assistant Professor with the Department of Computer Science, UWA. His research in computer vision is regularly published in the reputed scientific sources of the field. Previously, he has served as a Research Fellow with UWA and Australian National University. His research interests include semantic segmentation of 3D point cloud data, adversarial deep learning, video captioning, and remote sensing. He also serves as an Associate Editor for IEEE Access.



MICHAEL WISE received the Ph.D. degree in electrical engineering from the University of New South Wales. He was a Lecturer with the School of Computer Science, The University of Sydney, where he created computer software for use in plagiarism detection until the discovery that his programs had a secondary use in gene sequence alignment prompted him to shift his research to bioinformatics. He was subsequently employed as a Senior Research Fellow with the Pembroke College, Cambridge. In 2004, he moved to The University of Western Australia, where he had a joint appointment with the School of Biomedical, Biomolecular and Chemical Sciences and the School of Computer Science and Software Engineering. Since July 2016, he has been solely with the School of Computer Science and Software Engineering. His research interests include computational biology, particularly microbial informatics (i.e., the application of computational techniques to better understand bacterial and other microbial life), machine learning, and data science.

AJMAL MIAN (Senior Member, IEEE) is currently a Professor of computer science with The University of Western Australia. His research interests include computer vision, deep learning, shape analysis, face recognition, human action recognition, and video analysis. He was a recipient of two prestigious national fellowships from the Australian Research Council and several awards, including the West Australian Early Career Scientist of the Year 2012, the Excellence in Research Supervision, the EH Thompson Award, the ASPIRE Professional Development Award, the Vice-Chancellors Mid-Career Award, the Outstanding Young Investigator Award, the Australasian Distinguished Dissertation Award, and various best paper awards. He is also an Associate Editor of IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Image Processing, and the *Pattern Recognition* Journal. He served as the General Chair for DICTA 2019 and ACCV 2018.