Campus3D: A Photogrammetry Point Cloud Benchmark for Hierarchical Understanding of Outdoor Scene Supplementary Document

Xinke Li[1], Chongshou Li[1,*], Zekun Tong[1], Andrew Lim[1], Junsong Yuan[2]
Yuwei Wu[1], Jing Tang[1], Raymond Huang[1]
[1]Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore
[2]Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY, USA
{xinke.li,zekuntong}@u.nus.edu,{iselc,isealim}@nus.edu.sg,jsyuan@buffalo.edu
ywwu@u.nus.edu,{isejtang,raymond.huang}@nus.edu.sg

ACM Reference Format:
Xinke Li[1], Chongshou Li[1,*], Zekun Tong[1], Andrew Lim[1], Junsong Yuan[2] and Yuwei Wu[1], Jing Tang[1], Raymond Huang[1]. 2020. Campus3D: A Photogrammetry Point Cloud Benchmark for Hierarchical Understanding of Outdoor Scene Supplementary Document. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3394171.3413661

1 DATASET

1.1 Dataset Comparison

Data annotation comparisons between Campus3D and other point cloud datasets are summarized by Table 1.

Table 1: Annotation Comparison between Campus3D and other point cloud datasets of real environments.

Dataset	Designed Task	Hierarchical	Instance #	Class #	Muti-label
ScanNet	Object classification Semantic segmentation CAD model retrieval	No	36,213	20	No
S3DIS	Object detection	No	-	13	No
NYUv2	Semantic segmentation	No	35,064	894	No
SemanticKITTI	Semantic segmentation Semantic scene completion	No	-	25 28	No
Semantic3D	Semantic segmentation	No	-	8	No
Paris-Lille-3D	Semantic segmentation Instance segmentation	No	2,479 50	9	No
Campus3D (Ours)	**Semantic segmentation Instance segmentation**	**Yes**	**2,530**	**24**	**Yes**

1.2 Data Acquisition

The Campus3D dataset was constructed by the technique of Structure from Motion with Multi-View Stereovision (SfM-MVS) [3]. Here we describe our workflow of getting it. Devices to capture imagery were DJI Phaton 4 Pro drones equipping cameras with a 1-inch 2 MP CMOS sensors, and the drone flight planning mobile apps used in our application were DJI GS Pro and Pix4D Capture. The SfM-MVS software was Pix4Dmapper. The removal of pedestrians and cars are done automatically by Pix4Dmapper. The keypoints of non-static objects with changing relative positions on densely overlapped images were discarded by SfM process in the software[3].

To collect the data, drones were flown over all areas and took images with exact GPS coordinates. And then points would be generated by photogrammetry processing and registration from captured images and coordinates. Figure 1 displays our overall workflow.

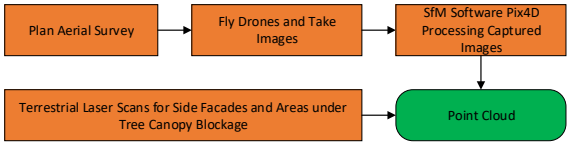


Figure 1: Workflow for point cloud generation.

The first step of workflow is to conduct the aerial survey including setting the flight routes and drone flying. We applied two types of flight routing strategies for the UAV photography: (1) grid and (2) circular, which were accessible in the drone flight planning mobile apps. Based on the height variance of objects in the targeted areas, the flight routing strategy was chosen such that the images were taken with the required overlap for the SfM software processing. The grid flight is suitable for most environments where the heights of buildings do not vary too much. In our image collection process, the height of the grid flight was set to be 10 meters - 15 meters above the highest building in the target area. The ground sampling distance (GSD) was programmed to be around 2cm with the highest of 1.63cm and the lowest of 3.48cm. Example camera positions from grid flight for a typical scene are illustrated by Figure 2 (a); the circular flight routing strategy is usually chosen for relatively high buildings in image capturing, where the drone flies an ellipse as shown in Figure 2 (b). This type of route guarantees that the images are taken from all angles around the center of the building. For extremely high buildings, we applied multiple circular flights at different heights. During the UAV image capturing, the drone was set as speed of 8m/s and flown when the clear view of image was guaranteed by weather.

After image collection on set flight route, the second step is to derive point clouds from images via SfM-MVS software, Pix4Dmapper. In this step, images with removal of error or blurry data were fed into Pix4Dmapper to perform matching based on the SIFT algorithm. From the initial matches, the Automatic Aerial Triangulation

*Corresponding author: Chongshou Li (iselc@nus.edu.sg).

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413661

(AAT) and Bundle Block Adjustment (BBA) were applied to generate a spare point cloud of feature tie points. Then we added ground control points correction which were measured with Real-Time Kinematics (RTK). Finally, multiple grid projections were combined by the tie points and the point cloud is produced.

In the end, we note that, for side facades and areas under tree canopy blockage, images are not able to be taken by drones. We complemented the aerial photogrammetry point cloud via tripod based terrestrial scanning around each building by FARO x330 scanner. The complementary point clouds were then registrated with origin data by two software, FARO Scene and Trimble Business Centre. This step makes sure that current point cloud dataset provides holistic views of constructions.

(a) Grid (b) Circular

Figure 2: Camera positions for grid and circular flight.

1.3 Coordinate System

There are three coordinate systems for presenting the location of the point data: (1) the SVY21 plane coordinate system with the origin of projection at (28001.642 m(E), 38744.572 m(N)) which is the raw data from GPS. (2) the campus coordinate system locating the origin point at the corner of the campus with raw coordinate (20774.967m(E), 30120.558m(W)). Comparing with the first system, the campus coordinate system provides axis value within a small scale. This results in memory reduction for processing the coordinates and make it easier to capture the relationship among regions and merge different regions. Therefore, users can easily change the size of training and test datasets according to the tasks' requirement; (3) local coordinate system of each region: this coordinate system with origins at the corner of each region can make the training set and test set independent with each other.

2 EXPERIMENT SETTING AND RESULTS

2.1 Experiment Setting of HL Method

The proposed method is based on PointNet++[1]. Here we present the parameter settings of it. The SA, FP and FC represent Set Abstraction Layer, Feature Propagation Layer and Fully Connected Layer respectively, the meaning of which are presented in [1].

- **Encoder**: SA(1024, 0.5, [32,32,64]) → SA(256, 1, [64,64,128]) → SA(64, 2, [128,128,256]) → SA(16, 4, [256,256,512])
- **Decoder**: FP(256, 256) → FP(256, 256) → FP(256, 128) → FP(128, 128, 128)
- **Classification head**: FC(128, 128) → FC(128, K_h), where K_h is number of labels in h_{th} level.

Moreover, we list the parameter setting of multitask loss. With definitions in the main paper,

- **Prediction loss**: $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 1$
- **Consistency loss**: $\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = \gamma_5 = 0.05$

Other parameter settings of training:

- **Loss Weights**: To deal with the imbalance label distribution, we applied $1/\log(N_c/N + t)$ as weights for each class where N_c is the points number of c_{th} class, N is the whole number of points, t is set as 1.2. **Optimizer**: Adam
- **Decay**: exponential decay with step 200000 and rate 0.7.
- **Epoch**: 150, **Sample Size**: 2048 points, **Minibatch Size**:16

For consistency loss (CL), we conducted analysis on various weight γ. Two observations were obtained: 1.As CL becomes more vital, the consistency rate grows and accuracy drops a lot. 2.Initial CL with large magnitude makes model only learn hierarchical relationships but not classification. For the sake of this, we applied a two-stage training policy for the proposed MT learning:

- **0-40 epoch**: training without consistency loss.
- **After 40 epoch**: training with consistency loss.

2.2 Results of Validation Set for HL Methods

Results of validation set for different HL methods are provided by Table 3 and Table 2.

Table 2: Validation results (class IoU%) for HL methods

Granularity Level	Class	Method				
		MC	MC+HE	MT_{nc}	MT	MT+HE
C^1	ground	87.3	**89.4**	89.3	89.3	**89.4**
	construction	64.2	**70.2**	69.8	70.1	70.0
C^2	natural	81.1	**82.1**	81.7	**82.1**	82.1
	man_made	46.8	48.6	47.3	**48.8**	**48.8**
	construction	68.6	**70.2**	68.4	70.0	70.0
C^3	natural	81.0	**82.1**	81.9	**82.0**	**82.1**
	play_field	2.0	9.5	13.8	**22.0**	20.7
	path&stair	2.9	2.8	3.4	0.0	0.0
	driving_road	41.3	44.2	43.4	**44.7**	44.5
	construction	69.1	**70.2**	69.0	70.0	70.0
C^4	natural	81.3	**82.1**	**82.1**	82.0	**82.1**
	play_field	14.8	9.5	12.6	20.6	**20.7**
	path&stair	**3.7**	2.8	3.5	0.0	0.0
	vehicle	42.6	45.2	44.0	**51.0**	**51.0**
	not vehicle	40.8	43.0	42.5	43.0	**43.2**
	building	66.6	69.3	68.7	70.0	**70.2**
	link	0.3	0.4	**0.9**	0.0	0.1
	facility	**0.1**	**0.1**	0.0	0.0	0.0
C^5	natural	81.7	**82.1**	82.0	81.9	**82.1**
	play_field	16.0	9.5	15.2	18.6	**20.7**
	sheltered	0.0	0.0	0.0	0.0	0.0
	unsheltered	1.4	1.0	**1.5**	0.0	0.0
	bus_stop	0.0	0.0	**0.7**	0.0	0.0
	car	46.4	47.4	47.6	54.4	**54.5**
	bus	5.9	**7.6**	2.3	0.7	0.7
	not vehicle	42.1	43.0	42.4	43.0	**43.2**
	wall	**50.6**	50.3	50.1	50.3	50.3
	roof	52.7	54.1	54.5	**55.1**	55.0
	link	0.7	0.4	**0.7**	0.1	0.1
	artificial_landscape	0.0	0.0	0.0	0.0	0.0
	lamp	0.0	0.0	0.0	0.0	0.0
	others	0.0	**0.1**	0.0	0.0	0.0

Table 3: Validation results (OA%) for different HL methods

Method	Granularity Level				
	C^1	C^2	C^3	C^4	C^5
MC	89.7	83.7	81.5	80.9	78.6
MC+HE	**91.5**	84.7	82.8	82.1	78.9
MT_{nc}	91.4	84.2	82.5	81.9	78.9
MT	**91.5**	**84.8**	**83.1**	**82.7**	79.5
MT+HE	**91.5**	**84.8**	**83.1**	**82.7**	**79.6**

2.3 Sampling Method

In this section, we present the pseudo-code of two sampling methods: (1) RC-KNN and (2) l-w RBS, in the following Algorithm 1 and Algorithm 2, respectively.

Algorithm 1: Random Center-KNN (RC-KNN) Sampling.

Global: voxel size d
Global: # of points per sample n, # of samples m

Input: points set S
Output: $\{S'_1, S'_2, \ldots, S'_m\}$

Initialization;
$k = n$
if *isTraining* **then**
  for $i = 1$ **to** m **do**
    $p_c \leftarrow$ `RandomSampling`$(S, 1)$
    $S'_i \leftarrow$ `QuerykNN`(S, p_c, k)
else
  forall p_c *of* `RandomVoxelSampling`(S, d, m) **do**
    $S'_i \leftarrow$ `QuerykNN`(S, p_c, k)

Input: point set S
Output: center points set S_{center}

Function `RandomVoxelSampling`(S, d, m):
  $S_{voxel} \leftarrow$ `VoxelSampling`(S, d)
  $S_{center} \leftarrow$ `RandomSampling`(S_{voxel}, m)

Algorithm 2: $l - w$ Random Block Sampling (l-w RBS).

Global: block size (l, w), overlap ratio r_o
Global: # of points per sample n, # of samples m

Input: points set S
Output: $\{S'_1, S'_2, \ldots, S'_m\}$

Initialization;
if *isTraining* **then**
  for $i = 1$ **to** m **do**
    $p_c \leftarrow$ `RandomSampling`$(S, 1)$
    $S_{block} \leftarrow$ `QueryBlock`(S, p_c)
    $S'_i \leftarrow$ `RandomSampling`(S_{block}, n)
else
  forall p_c ***of*** `TravelsalCenter`(S, r_o) **do**
    $S_{block} \leftarrow$ `QueryBlock`(S, p_c)
    $S'_i \leftarrow$ `RandomSampling`(S_{block}, n)

Input: point set S, center point p_c
Output: block points set S_{block}

Function `QueryBlock`(S, p_c):
  $S_{block} = \emptyset$
  $(x_1^c, x_2^c, x_3^c) = p_c$
  forall (x_1^i, x_2^i, x_3^i) ***of*** S **do**
    if $x_1^i \in [x_1^c - \frac{l}{2}, x_1^c + \frac{l}{2}]$ ***and*** $x_2^i \in [x_2^c - \frac{w}{2}, x_2^c + \frac{w}{2}]$ **then**
      $S_{block} \leftarrow S_{block} \cup \{(x_1^i, x_2^i, x_3^i)\}$
  return S_{block}

2.4 Sampling Parameters

For RBS method, the essential setting is the block size. To investigate the setting of this parameter, we counted the number of points for different block sizes and presented the 20/80 percentile of points number in Figure 3.

Given that the fixed input size of models is 2048, we choose 12m x 12m as block size since (1) blocks in this size with small points number and/or sparse points distribution still contain sufficient points, and (2) blocks in this size with dense points distribution and/or large points number are not overfull and do not suffer severe information loss from the sampling.

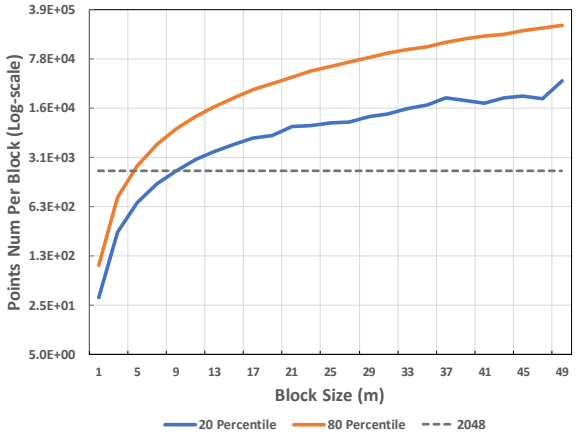


Figure 3: Points number per block for different block sizes.

2.5 Evaluation Metric for Benchmark

The benchmark applied two common metrics for evaluation of the point cloud semantic segmentation task: mean Intersection-of-Union (mIoU) and Overall Accuracy (OA). Here we provide their definitions. Let M be an $N \times N$ confusion matrix of the chosen classification method, where each entry m_{ij} is a number of samples from ground-truth class i predicted as class j. In terms of semantic segmentation, the definition of OA is eq. (1),

$$\mathrm{OA} = \frac{\sum_{i=1}^{N} m_{ii}}{\sum_{j=1}^{N} \sum_{k=1}^{N} m_{jk}}, \tag{1}$$

which mainly evaluates accuracy of point-wise predictions. And mIoU is defined in eq. (2), where IoU_i is the IoU for ith class.

$$\mathrm{mIoU} = \frac{\sum_{i=1}^{N} \mathrm{IoU}_i}{N} \text{ where } \mathrm{IoU}_i = \frac{m_{ii}}{m_{ii} + \sum_{j \neq i} m_{ij} + \sum_{k \neq i} m_{ki}} \tag{2}$$

For instance segmentation, we applied coverage (Cov) and weighted coverage (WCov) introduced into 3D tasks by Wang and Jia [2], with the definition as eq. (3), where $r^{\mathcal{G}}$ and r^{O} are ground-truth and predict grouping points respectively.

$$\begin{aligned} \mathrm{Cov}(\mathcal{G}, O) &= \sum_{i=1}^{|\mathcal{G}|} \frac{1}{|\mathcal{G}|} \max_j \mathrm{IoU}\left(r_i^G, r_j^O\right) \\ \mathrm{wCov}(\mathcal{G}, O) &= \sum_{i=1}^{|\mathcal{G}|} w_i \max_j \mathrm{IoU}\left(r_i^G, r_j^O\right) \\ w_i &= \frac{|r_i^G|}{\sum_k |r_k^G|} \end{aligned} \tag{3}$$

3 DATA STATISTICS

3.1 Region Full Name

Here we provide the full name for the six regions in Table 4. Each region is a complete outdoor scene in the NUS campus.

Table 4: Full Names of Regions.

Name	Full Name
FASS	Faculty of Arts and Social Sciences
FOE	Faculty of Engineering
PGP	Prince George's park Residences
YIH	Yusof Ishak House
UCC	University Culture Centre
RA	Ridge Area

3.2 Instance Distribution

Table 5 displays the instance distribution across regions and classes, where "*vehicle*" in the category of "*driving_road*", "*roof*" and "*wall*"in the category of "*building*" take the majority of number . We note that the small number of instances of "*others*" is defined by the class label's definition. And it is hard to identify plants instances in "*natural*", because it is connected with each other and covers at least 30% area at each region (see. Figure 3).

Table 5: Number of instances in each class and region

Labels	**Area**						
	FASS	FOE	PGP	RA	UCC	YIH	**Total**
Unclassified	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
Ground	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
Construction	42	63	64	75	11	37	292
Man-made (ground)	55	25	39	52	22	56	249
Natural	1	1	1	1	1	1	6
Play field	6	2	3	2	0	1	14
Path&stair	46	21	13	47	13	49	189
Driving Road	3	2	23	3	9	6	46
Buidling	33	31	58	47	8	20	197
Link	3	20	6	9	3	11	52
Facility	6	12	0	19	0	6	43
Sheltered (path)	1	6	0	2	3	9	21
Unsheltered (path)	40	12	12	22	8	40	134
Bus stop	5	3	1	23	2	0	34
Vehicle(Driving road)	391	143	107	192	74	107	1014
Without vehicle(Driving road)	3	2	23	3	9	6	46
Car	374	143	103	188	73	107	988
Bus	17	0	4	4	1	0	26
Wall	33	31	58	47	8	20	197
Roof	146	165	229	152	52	140	884
Artificial landscape	2	1	0	0	0	3	6
Lamp	4	10	0	10	0	3	27
others	0	1	0	9	0	0	10

3.3 Leaf Node Class Distribution Among Regions

For the class of leaf node in the label tree, a pie chart summary of each region is provided by Figure 4. It demonstrates significant differences among regions in terms of class composition. This property is good for constructing training sets with strong generalization ability and comprehensive test sets with uniqueness of features.

3.4 Dataset Visualization

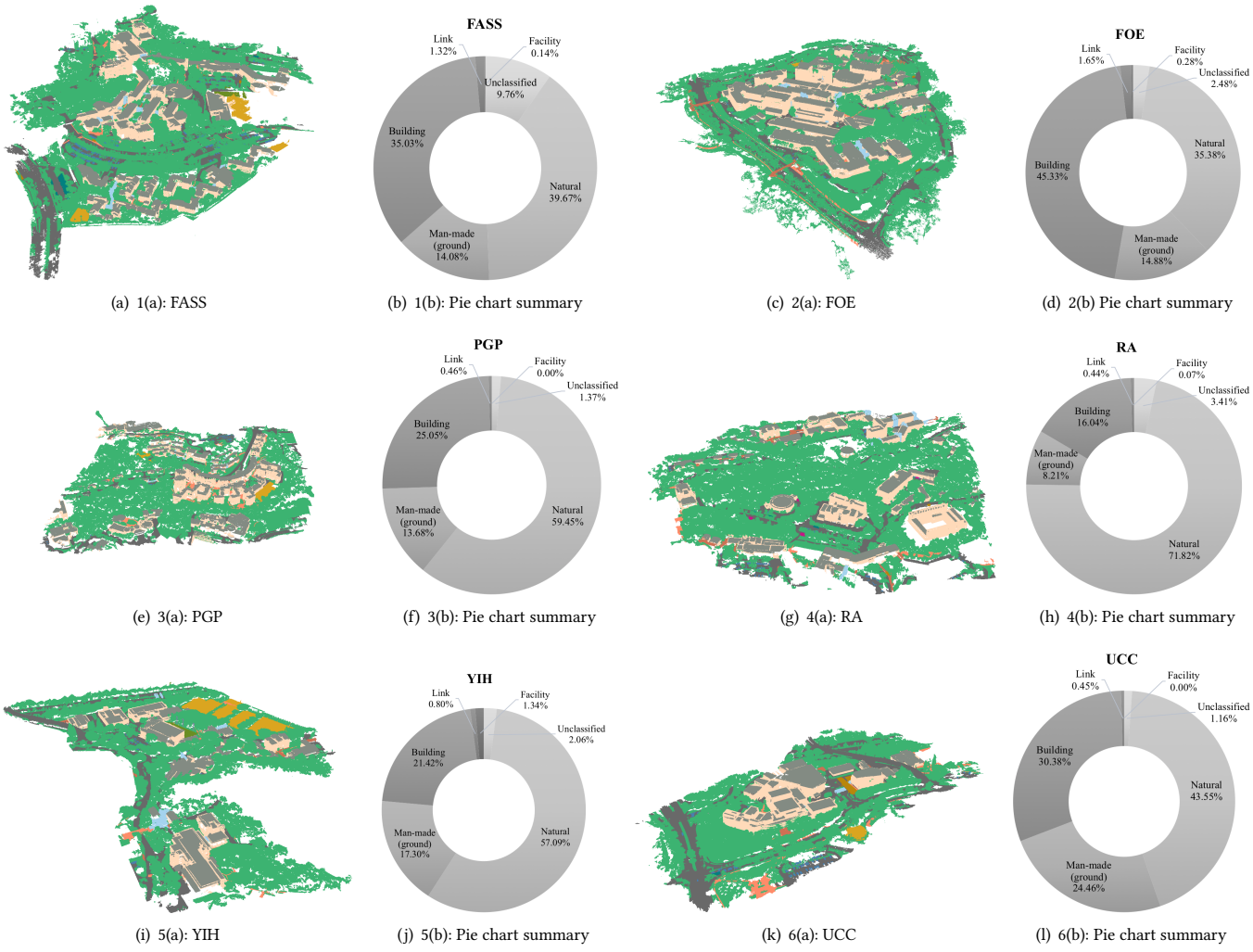
In the end, we provide additional visualization for six regions' scene covered by the Campus3D in Figure 5.

REFERENCES

[1] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*. 5099–5108.

[2] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. 2019. Associatively segmenting instances and semantics in point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4096–4105.

[3] Matthew J Westoby, James Brasington, Niel F Glasser, Michael J Hambrey, and Jennifer M Reynolds. 2012. 'Structure-from-Motion'photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology* 179 (2012), 300–314.

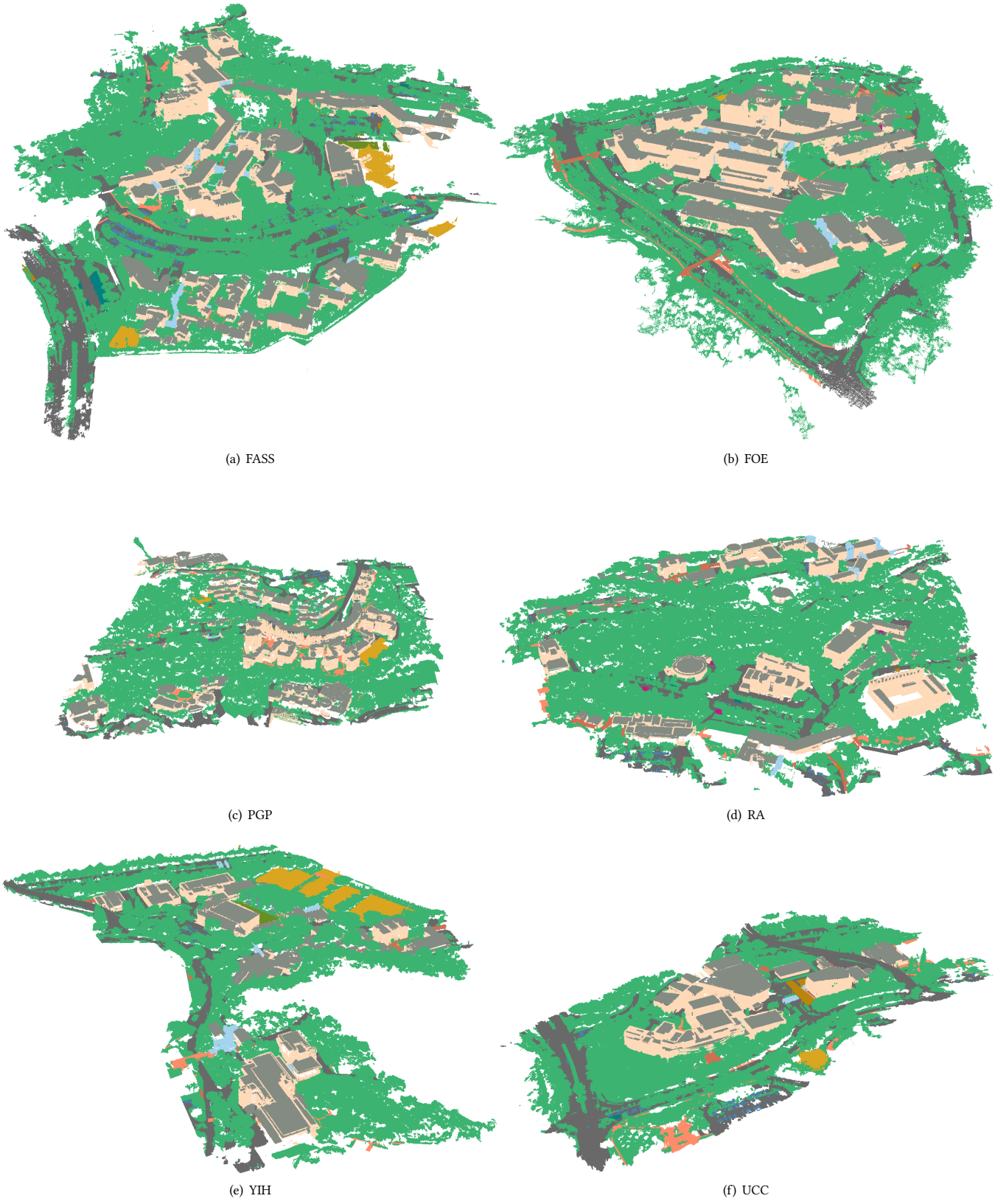


(a) 1(a): FASS (b) 1(b): Pie chart summary (c) 2(a): FOE (d) 2(b) Pie chart summary

(e) 3(a): PGP (f) 3(b): Pie chart summary (g) 4(a): RA (h) 4(b): Pie chart summary

(i) 5(a): YIH (j) 5(b): Pie chart summary (k) 6(a): UCC (l) 6(b): Pie chart summary

Figure 4: Visualization and pie chart summary of leaf node label's distribution in the six regions.



(a) FASS

(b) FOE

(c) PGP

(d) RA

(e) YIH

(f) UCC

Figure 5: Visualization of the six regions.