Appendices

A Data availability

Our STPLS3D datasets can be downloaded at: www.stpls3d.com/data

B Data collection & generation details

Video Illustrations. We provide an video demo illustrating our synthetic data generation pipeline discussed in Section 3. The video can be viewed at https://youtu.be/6wYWVo6Cmfs.

Real-World Datasets To capture the real-world 3D data, we first used DJI Phantom 4 Pro for the collection of aerial images. An autonomous UAV-path planning and imagery collection system called rapid aerial photogrammetric reconstruction system (RAPTRS) [69] was adopted for the survey. Specifically, RAPTRS encodes photogrammetry best practices and allows aerial photographs collected with multiple flights to cover a large area of interest. The aerial images were collected using a crosshatch-type flight pattern with predefined overlaps ranging from 75%∼85% and flight altitudes ranging from 25m∼70m. We conducted surveys on four real-world sites, including the University of Southern California Park Campus (USC), Wrigley Marine Science Center (WMSC) located on Catalina Island, Orange County Convention Center (OCCC), and a residential area (RA).

For richer diversity, we selected the area of interest to have different building and terrain types. USC campus mainly consists of commercial buildings with the paved ground (vehicle roads, pedestrian roads, and squares). Approximately 20% of the campus is covered by grassland and tree canopy. The average height of buildings is around 5∼6 floors. WMSC contains terrain with a valley and cliffs located on the shoreline of an island. OCCC is a large convention center located in the tourist district of Orlando. The surroundings of OCCC, including the parking lots and vegetated areas, are also included in our datasets. The residential area (RA) covers a typical American residential area that contains single and townhouses with an average height of 1∼2 floors.

All 3D point clouds are reconstructed using the commercial software – *i.e.*, ContextCapture. Each point is enriched with one of the six semantic class labels, including *ground*, *man-*

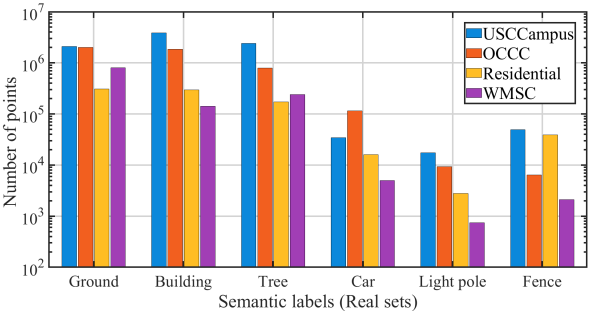

Figure 4: The class distribution of *real-dataset* of our STPLS3D. Note the logarithmic scale for the vertical axis.

made structures (including outdoor furniture, construction equipment, site storage trailers, *etc.*), *trees*, *cars*, *light poles*, and *fences*. Note that the raw point clouds are subsampled to 0.3 m point spacing for training and evaluating existing segmentation methods. The statistics of the real-world dataset are also provided in Figure 4.

Synthetic Datasets Following the proposed data generation pipeline, we further generated a large-scale synthetic dataset with various landscapes (*i.e.*, various terrain shapes, types of vegetation, and urban density). It is composed of 62 point clouds and covers approximately 16 square kilometers of landscape. Additionally, the flight altitudes were set in the range of 60m∼120m. Image overlaps were set in the range of 75%∼85%, and sunlight directions were randomly assigned to simulate the data collection at different times in a day. In particular, three versions of the synthetic datasets were generated with different focuses.

SyntheticV1. This dataset was created using limited game objects and composed of 7 semantic categories, including *ground*, *building*, *vegetation*, *vehicle*, *light pole*, *street sign*, and *clutter*. In this version, we mainly focused on adding diversity for large objects. For instance, details were added to the ground by randomly sculpting the input digital surface model with simulated ditches, street gutters, speed bumps, *etc*. A large number of forests were also added by placing trees in random polygons with different tree spacing.

SyntheticV2. In this version, we focused on adding diversity for the terrain shapes and visual appearances, as well as the variation of the small objects. In particular, we selected DSM with large slopes, enriched the game objects repository, and expanded the fine-grained semantic categories. *Low* (0.5m <height<=2.0m), *medium* (2.0m<height<=5.0m) and *high* (5.0m<height) vegetation class labels were adopted to separate different kinds of vegetation following ASPRS specification [2]. Vehicles were further divided into passenger cars (including sedan and hatchback cars) and trucks. *Bikes*, *motorcycles*, *fences*, *roads*, *aircraft*, and *military vehicles* were also incorporated into the 3D scene generation process. A procedural landscape material was also leveraged to automatically generate grass and rocky textures based on the ground slope. The contextual relationship between objects was also considered, where *vehicles* were placed on the *roads* and *light poles*, and *street signs* were placed alongside the *roads*. Finally, it is noted that the instance annotations for specific objects (*e.g.*, *cars, trees, buildings, bikes, etc.*) were also introduced in this version.

SyntheticV3. In this version, we focused on large size building footprints to simulate urban areas and increase the variation of the object materials. A database of materials (including metal, rubber, signs, car paints, *etc.*) for small objects (such as *vehicles*, *light poles*, *street signs*, *bikes*, *motorcycles*, *etc.*) was created. These materials were assigned to each object during generation. We also exploited the off-the-shelf library of photogrammetry-based textures – *i.e.*, Quixel Megascans for changing materials of buildings and fences. Considering that simply assigning random materials to the facade of the building may reduce the realism of the 3D environment, we first assigned the material categories (*e.g.*, brick, concrete, wood, *etc.*) to different building components (*e.g.*, wall, roof, *etc.*). Next, individual material was randomly selected for each building component from the given material category. In particular, two new ground material labels (*i.e.*, grass and dirt) were also introduced in this version. Dirt texture was painted around the building footprints with a predefined buffer, and grass texture was used to fill the blank areas that did not belong to dirt or road. The statistics of different synthetic data are shown in Figure 5. we provide additional visualization of our synthetic and real-world datasets in Figure 6.

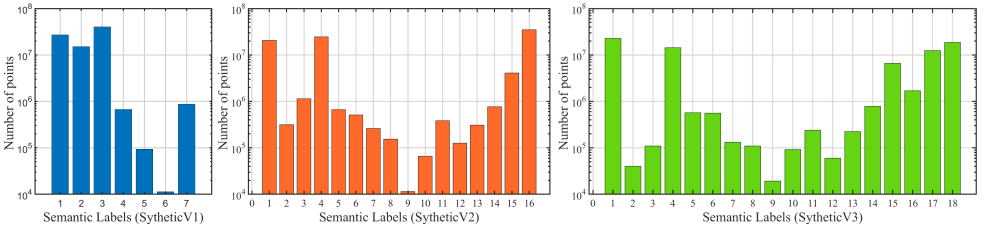

Figure 5: The class distribution of *synthetic* subsets of our STPLS3D. Note the logarithmic scale for the vertical axis.

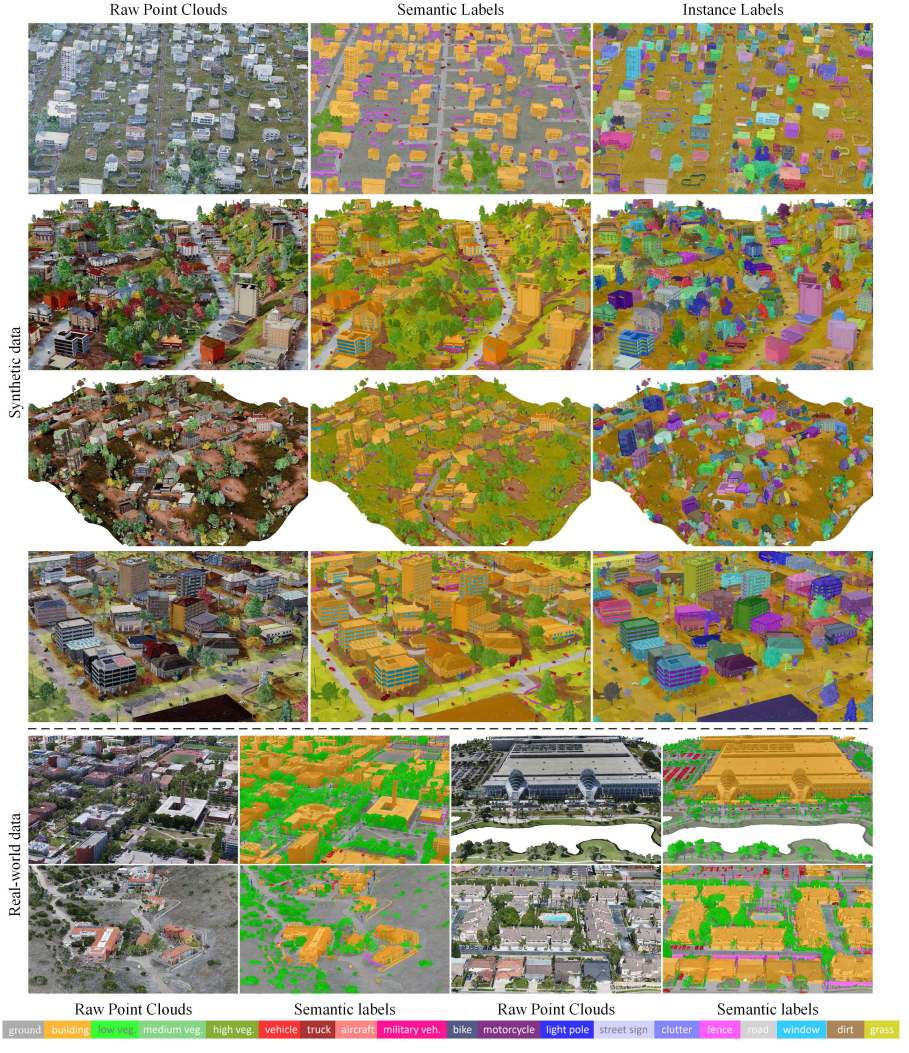

Figure 6: Examples of synthetic and real-world point clouds in our STPLS3D dataset. Different semantic classes are shown in different colors, as illustrated in the color legend. Note that different instances are displayed in different random colors. Best viewed in color.

C Comparison of Data Quality

As discussed in Section 3.2, directly sampling or ray casting point clouds from the 3D virtual environment will lead to a large domain gap from the real photogrammetry data. Here, we provide an intuitive comparison by visualizing the ray casting 3D points, the synthetic photogrammetric points, and real-world photogrammetric points of tree crowns in Figure 7. Additionally, the sectional view and volume density histogram is also reported.

It can be seen that: 1) From the sectional view, it is clear that points of the synthetic ray-casted point clouds are scattered inside the tree crowns, while synthetic and real photogrammetry point clouds exhibit hollow-shaped shells. 2) The volume density histograms show that synthetic and real photogrammetry point clouds have much more similar point distributions compared with the synthetic ray-casted point clouds. Overall, the point clouds generated from our synthetic photogrammetry pipeline are closer to the real data. For comprehensive visualization, we also provide an anonymous video demo demonstrating the quality and distribution of different point clouds generated by ray casting, synthetical photogrammetry, and real photogrammetry. The video can be viewed at https://youtu.be/4AjMWTgV2Ec.

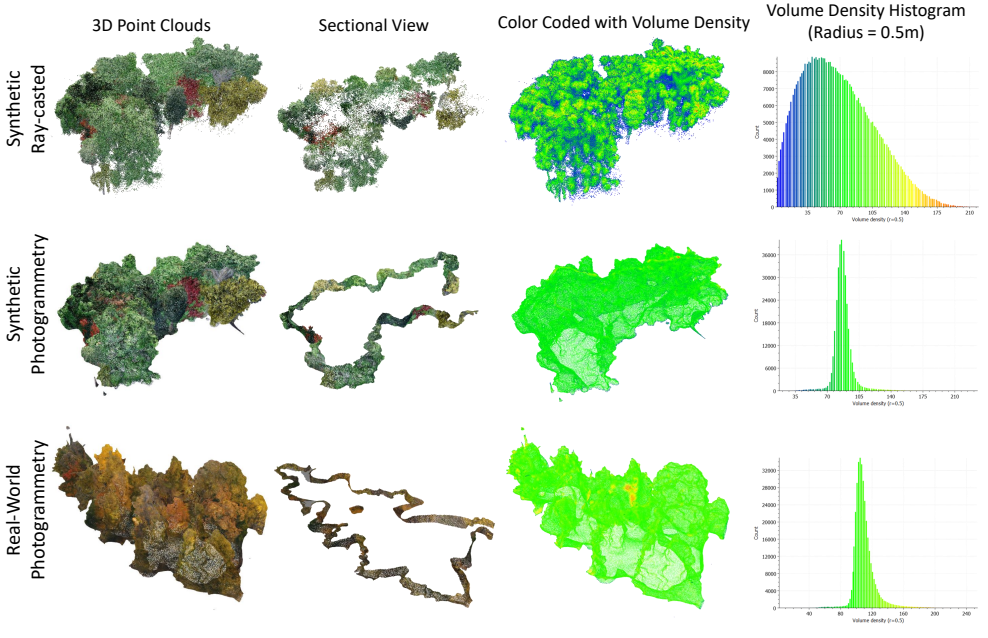

Figure 7: Qualitative comparison of tree crowns generated by ray-casted, synthetic photogrammetry, and real photogrammetry.

D Object placement principles for creating the synthetic environment layouts

In this study, we empirically developed several simple yet effective parameterized scene layout principles. 1) Placing vehicles, trucks, bikes, and motorcycles on the paved ground and road based on the road width length with randomized intervals. 2) Placing city furniture alongside the roads with parameterized buffer sizes. 3) Placing clusters of bikes, motorcycles, and small objects around buildings and fences. 4) Scatter polygons and rings with random shapes and sizes to cover a parameterized percentage of the empty space and place objects within the same category in each polygon and ring with the parameterized minimum allowed distance. 5) Randomize the objects' rotations and scales. By mixing and matching these simple rules with parameters randomly sampled within reasonable ranges, unique strategies can be created to produce scene layouts with a large diversity and a certain realism.

E Evaluation of the synthetic V1, V2, and V3

Table 5: Quantitative evaluation of different versions of the STPLS3D synthetic datasets using real subsets as testing cases. Overall Accuracy (oAcc, %) and mean IoU (mIoU, %) are reported.

	RandLA-Net [35]		SCF-Net [22]		KPConv [72]	
	oAcc(%)	**mIoU(%)**	oAcc(%)	**mIoU(%)**	oAcc(%)	**mIoU(%)**
V1	78.51	44.42	76.40	43.85	76.71	44.36
V2	80.86	46.67	80.46	46.77	84.18	53.88
V3	76.95	47.97	76.43	47.38	73.42	48.41
V12	86.25	50.41	86.31	53.37	85.75	51.43
V13	80.41	49.08	79.48	49.66	79.73	51.08
V23	84.32	51.07	85.41	52.15	85.45	55.82
V123	**86.39**	**55.76**	**87.86**	**56.60**	**88.04**	**58.05**

In Section 5.1, we have evaluated the segmentation performance of baseline networks on our real-world dataset by training on the synthetical V1-V3 dataset. To further determine the impact of different versions of the synthetic subset on the final segmentation performance, we conducted the 7 groups of experiments as follows. Note that all groups of experiments were tested on the real-world dataset but trained with synthetic datasets with different settings.

- Trained in synthetic V1 only.
- Trained in synthetic V2 only.
- Trained in synthetic V3 only.
- Trained in synthetic V1+V2.
- Trained in synthetic V1+V3.
- Trained in synthetic V2+V3.
- Trained in all the synthetic V1+V2+V3.

The quantitative performance evaluation of three baselines is shown in Table 5. It can be seen that: 1) The overall performance of all baselines improved when training with the combinations of synthetic subsets, compared with training on the individual synthetic subset. 2)

All baselines achieved the best performance when trained on all synthetic subsets, indicating the positive impact of the synthetic datasets, especially when the real-world training data is scarce or difficult to acquire.

F Evaluation on Synthetic V3 with 18 labels

Table 6: Quantitative results on the synthetic v3 subset.

	mIoU(%)	Build.	LowVeg.	MediumVeg.	HighVeg.	Vehicle	Truck	Aircraft	MilitaryVeh.	Bike	Motorcycle	LightPole	StreetSign	Clutter	Fence	Road	Windows	Dirt	Grass
RandLA-Net [🔲]	67.52	94.11	39.42	46.84	96.32	82.39	82.45	66.72	70.02	21.72	56.54	78.27	40.22	55.25	78.50	80.64	59.28	80.37	**86.22**
SCF-Net [🔲]	69.07	93.31	**45.40**	**49.40**	96.51	82.17	83.08	68.28	71.93	22.29	**57.98**	80.99	**45.27**	**63.06**	79.29	79.97	57.63	**80.79**	85.86
KPConv [🔲]	**70.35**	**96.18**	35.17	47.88	**97.04**	**86.44**	**84.24**	**75.35**	**72.26**	**23.09**	57.59	**86.65**	43.07	62.69	**85.60**	**81.82**	**67.23**	79.29	84.68

Here, we evaluated the segmentation performance of three baseline networks (*i.e.*, KPConv, RandLA-Net, SCF-Net) on the synthetic subset of our dataset. In light of the data diversity and quality, we selected the SyntheticV3 dataset for evaluation. The quantitative results achieved by different baselines are shown in Table 6. It can be seen that KPConv achieved the best segmentation performance on the SyntheticV3 subset, with a mIoU score of 70.35%, followed by the SCF-Net and RandLA-Net. We also noticed that all three baselines failed to achieve satisfactory performance on small objects such as low vegetation, bikes, and street signs. This is likely because the geometry and texture details of these small objects were lost during the 3D reconstruction in the photogrammetry process. Additionally, the performance on the underrepresented categories such as *medium vegetation* is also far from satisfactory, indicating learning from imbalanced class distribution remains a challenging problem for existing techniques.

G Instance Segmentation with Reduced Semantic Classes

Table 7: Quantitative evaluation of two instance segmentation baselines on the synthetic v3 subset with reduced semantic classes.

	Metric	mean (%)	Build.	Vege.	Vehicle	Large Vehicle	Aircraft	Bike	Poles & Signs	Clutter	Fence
HAIS[16]	AP	42.9	68.1	22.1	77.1	48.9	47.0	46.8	26.1	24.2	25.9
	AP50	54.6	74.1	27.2	87.2	57.8	67.5	67.0	34.1	29.5	47.0
	AP25	60.1	75.7	29.9	89.1	62.5	71.1	73.9	35.9	32.5	70.1
PointGroup[41]	AP	33.4	63.6	19.8	57.0	44.4	36.9	20.0	21.7	18.2	19.6
	AP50	52.0	71.8	26.1	83.9	59.7	66.1	51.5	41.4	25.0	42.8
	AP25	61.0	75.4	30.5	87.0	64.2	71.6	70.5	54.7	28.5	66.7

Since the instance segmentation performance depends on the quality of the semantic segmentation results, We also provided an instance segmentation benchmark with reduced

semantics by merging similar semantic classes to eliminate the cascade effect from poor semantic segmentation. Specifically, *low*, *medium*, and *high vegetation* were merged into the *vegetation* category. *Bicycle* and *motorcycle* points were labeled as *bike*. *Trucks* and *military vehicles* were combined as *large vehicles*. The *street signs* were joined with *light poles*. Thus, the semantics were reduced from 14 classes to 9 classes. PointGroup [41] and HAIS [16] were used again as the baselines. The results are shown in Table 7. Both HAIS and PointGroup achieved a higher AP (nearly 9%), which shows that if the semantic segmentation capability for similar object classes could be improved in the end-to-end instance segmentation networks, the performance of instance segmentation could also be increased.

H Data-Related Hyperparameters of Benchmark Methods

To achieve a trade-off between data scale, resolution, and computing resources, we empirically set 0.3m for grid downsampling to reduce the number of total points while preserving enough details. In addition, for voxel-based approaches, including MinkowskiNet, PointGroup, and HAIS, we set the sample size of $50\,\text{m} \times 50\,\text{m}$ on the XY plane. We used a sample size of $100\,\text{m} \times 100\,\text{m}$ for PointTransformer, an 18m radius of sphere for KpConv, and 40,960 input points for SCF-Net and RandLA-Net.

I Definition of Semantic Categories

Here, we provide a detailed definition of the semantic categories in our STPLS3D dataset.

SyntheticV1:

1. Ground: including grass, paved road, dirt, etc.
2. Building: including commercial, residential, educational buildings.
3. Vegetation: including low, medium, and high vegetation.
4. Vehicle: including sedan and hatchback cars.
5. Light pole: including light poles and traffic lights.
6. Street sign: including road signs at the side of roads.
7. Clutter: including city furniture, construction equipment, barricades, and other 3D shapes.

SyntheticV2:

1. Building: Same as the definition of building in SyntheticV1.
2. Low vegetation: 0.5 m $<$ vegetation height $<$ 2.0 m.
3. Medium vegetation: 2.0 m $<$ vegetation height $<$ 5.0 m.
4. High vegetation: 5.0 m $<$ vegetation height.
5. Passenger car: including sedans and hatchback cars.
6. Truck: including pickup trucks, cement trucks, flat-bed trailers, trailer trucks, etc.
7. Aircraft: including helicopters and airplanes
8. Military vehicle: including tanks and Humvees.
9. Bike: bicycles.
10. Motorcycle: motorcycles.

11. Light pole: Same as the definition of light pole in SyntheticV1.
12. Street sign: Same as the definition of street sign in SyntheticV1.
13. Clutter: Same as the definition of clutter in SyntheticV1.
14. Fence: including timber, brick, concrete, metal fences.
15. Road: including asphalt and concrete roads.
16. Grass: including grass lawn, wild grass, etc.

SyntheticV3:

16. Window: glass windows.
17. Dirt: bare earth.
18. Grass: Same as the definition of grass in SyntheticV2.

Note that, the definition of classes 1 to 15 is the same as SyntheticV2.

Real-world data:

1. Ground: including grass, paved roads, dirt, sidewalk, parking lots, etc.
2. Tree: including low, medium, and high vegetation.
3. Car: including sedans and hatchback cars, pickup trucks, flatbed trailers, trailer trucks, etc.
4. Light pole: including light poles, traffic lights, and street signs.
5. Fence: including timber, brick, concrete, metal fences.
6. Building (man-made structure): Including buildings, city furniture, construction equipment, site storage trailers, *etc.* (*i.e.*, Objects that do not belong to ground, tree, car, light pole, and fence.)

J Class mapping between synthetic and real data

Considering that the semantic categories of the synthetic datasets are inconsistent with the real-world dataset (18 vs. 6), we conducted a class mapping to unify the semantic categories for the experiments discussed in 5.1. Specifically, *road*, *dirt*, and *grass* points were combined as *ground*. *Low*, *medium* and *high vegetation* were merged into the *vegetation* category. *Cars*, *trucks*, and *military vehicles* were labeled as *vehicles*. The *street sign* was joined with *light poles*, and all other objects except fences were merged with buildings as *man-made structures*.

K Visualization of the FDc dataset

To have an intuitive and clear understanding of the FDc data, we visualize the 3D point cloud along with its annotations in Figure 8.

Figure 8: Example visualization of the FDc dataset.