AN EFFICIENT SEMI-AUTOMATED SCHEME FOR LIDAR ANNOTATION AND A BENCHMARK INFRASTRUCTURE DATASET

Aotian Wu, Pan He*, Xiao Li, Ke Chen, Sanjay Ranka, Anand Rangaragan
Department of Computer and Information Science and Engineering
The University of Florida, Gainesville, Florida, USA
{aotian.wu, pan.he, xiao.li, ke.chen, ranka, anand}@ufl.edu

ABSTRACT

We present an efficient semi-automated annotation tool that automatically annotates LiDAR sequences with tracking algorithms while offering a fully annotated infrastructure LiDAR dataset—FLORIDA (Florida LiDAR-based Object Recognition and Intelligent Data Annotation)—which will be made publicly available. Our annotation tool seamlessly integrates multi-object tracking (MOT), single-object tracking (SOT), and batch editing functionalities. Specifically, we introduce a human-in-the-loop schema where annotations are incrementally added to the training set of MOT and SOT models after being fixed and improved by human annotators. By repeating the process, we significantly increase the overall annotation speed by $3-4$ times and obtain higher quality annotations than a state-of-the-art annotation tool. The human annotation experiments verify the effectiveness of our annotation tool. In addition, we provide detailed statistics and object detection evaluation results for our benchmark dataset at a busy traffic intersection.

1 INTRODUCTION

Most existing perception systems rely on sensory data acquired from cameras, which perform poorly in low light and adverse weather conditions. To resolve this limitation, we have witnessed advanced LiDAR sensors become popular in perception tasks in autonomous driving applications. Nevertheless, their usage in traffic monitoring systems is less ubiquitous. We identify two significant obstacles in cost-effectively and efficiently developing such a LiDAR-based traffic monitoring system: (i) public LiDAR datasets such as Waymo Sun et al. (2020) are insufficient for supporting perception tasks in infrastructure systems as they have different viewpoints and only provide a limited set of pedestrians, and (ii) 3D annotations on LiDAR point clouds are time-consuming and expensive. For example, as stated in Cong et al. (2022), it took 960 person-hours effort of 20 professional annotators to annotate 219K bounding boxes in the point clouds.

To fill this gap, we present an efficient semi-automated annotation tool that automatically annotates LiDAR sequences with human-in-the-loop initialization and correction. In this work, we construct a fully annotated infrastructure LiDAR dataset that

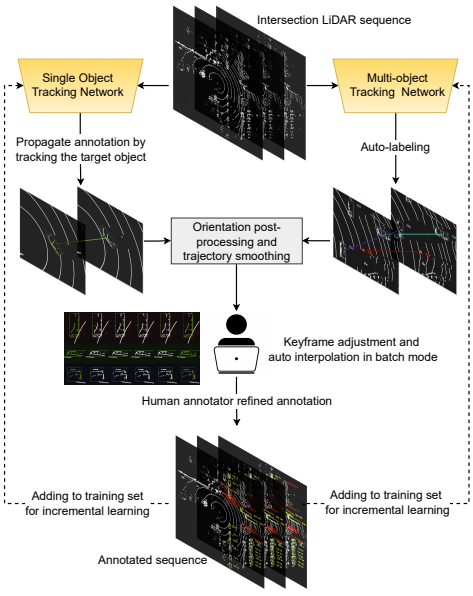


Figure 1: Overview of the semi-automated annotation pipeline.

*indicates corresponding author

Table 1: Comparison of FLORIDA with other popular infrastructure lidar benchmarks.

Dataset	With crowded pedestrians	Include all traffic participants	Release full dataset with labels	Object detection evaluation	Annotation method
Ko-PER	✗	✓	✓	✗	Not mentioned
PedX	✓	✗	✓	✗	3D model fitting for auto 3D labeling from 2D segmentation and joint location labels
IPS 300+	✓	✓	✗	✓	By Datatang Co. Ltd.
LUMPI	✓	✓	✗	✗	1) Foreground segmentation using DBSCAN algorithm 2) Annotation propagation using Kalman Filter 3) 3D pose correction using ICP 4) Costly human refinement
FLORIDA (Ours)	✓	✓	✓	✓	1) Auto labeling using MOT 2) Missing object annotation using SOT 3) Pedestrian Orientation auto-correction from moving direction 4) Human refinement in batch mode

will be made publicly available. We seek to use Single Object Tracking (SOT)—a deep learning-based object tracking algorithm—for annotation propagation. Given an object's first-frame annotation, our algorithm can track it robustly in the subsequent frames while maintaining the flexibility of being trained on autonomous driving LiDAR datasets or infrastructure LiDAR datasets. Through extensive experiments, we find that it works well in practice. Furthermore, inspired by the work Qi et al. (2021), we incorporated a Multi-Object Tracking (MOT) algorithm into our annotation tool. Unlike the SOT, which focuses on independently annotating and refining each object instance via labeling the first frame of each object and propagating it to subsequent frames, the MOT algorithm can automatically detect and track all object instances of a scene in a single shot. Once it generates the predicted annotation, human annotators may visually inspect and adjust the results. In practice, initial annotations are provided by a trained MOT model. If MOT fails to detect an object, one can annotate its first appearance and utilize a SOT model to propagate. Both SOT and MOT models may not initially give desirable predictions for annotation. Our human-in-the-loop schema allows us to fix and refine imperfectly predicted annotations and improve upon them to recursively obtain better annotations. We show through experiments that the model prediction accuracy is consistently enhanced by adding more high-quality annotations to the training set.

2 BACKGROUND

3D Single Object Tracking on Point Clouds. In 2019, SC3D Giancola et al. (2019) introduced the 3D SOT problem and implemented a Siamese tracker that encodes the target and candidates into embeddings, followed by the cosine similarity measure to determine the best-matching candidate. P2B Qi et al. (2020) argued that SC3D's candidate generation is either time-consuming or performance-degraded. It then proposed an end-to-end Siamese tracker by customizing Deep Hough voting Qi et al. (2019). Multiple successive works Zheng et al. (2021); Wang et al. (2021b); Hui et al. (2021); Zhou et al. (2022) are based on P2B with additional innovations w.r.t. feature extraction, template and search area feature fusion, and detector heads. Recently, M^2-track Zheng et al. (2022) proposed the motion-centric paradigm and achieved state-of-the-art performance on multiple benchmarks. In this paper, we adopt M^2-track as the SOT model in our annotation tool.

3D Multi-Object Tracking on Point Clouds. The research community initially analyzed the MOT problem in 2D image-based representations. For 2D MOT, data association is based on appearance and motion cues. For 3D MOT in point clouds, appearance cues become less discriminative due to the data sparsity and lack of texture information. Therefore, motion cues are more reliable because the scale of an object remains constant, and there are no abrupt movements. Most 3D MOT models employ the tracking-by-detection paradigm. For example, AB3DMOT Weng et al. (2020) proposed a baseline approach that adopts the 3D Kalman Filter for motion models and the Hungarian algorithm for data association. Follow-up work Wu et al. (2021); Chiu et al. (2021) mainly improves upon its data association and life cycle management strategy. SimpleTrack Pang et al. (2021) encapsulates multiple 3D MOT methods into a unified framework. We employ SimpleTrack due to its flexibility and simplicity.

Smart Annotation Tools for Point Clouds. 3D BAT Zimmer et al. (2019) is one of the earliest open-sourced point cloud annotation tools—a web-based application with multi-model data. The annotations for point clouds are automatically projected to different camera views. It also supports interpolation between keyframes to accelerate sequence annotation. LATTE Wang et al. (2019)

further implemented sensor fusion, smart one-click annotation, and integrated tracking into sequence annotation. LATTE used a clustering algorithm to achieve the one-click annotation to find all points for the target object, estimate the 2D bounding box (BBox), and convert to 3D BBox coordinates based on camera-LIDAR calibration. In addition, LATTE utilized the Kalman Filter algorithm for tracking objects. SAnE Arief et al. (2020) improves one-click annotation by employing a denoising pointwise segmentation strategy that assigns a noise penalty for all boundary locations to better separate nearby objects. SAnE also proposed an improved tracking algorithm.

SUSTechPOINTS Li et al. (2020) is one of the best open-source point cloud annotation tools to the best of our knowledge. It has a handy interface for adjusting BBox in single frame or batch mode. Moreover, it implements a collection of functions, such as one-click annotation and annotation propagation. It employs a heuristic registration algorithm that calculates the relative geometric transformation between the target in consecutive frames to propagate the current BBox to subsequent frames. Unfortunately, the registration performance is imperfect, requiring a certain amount of effort in label refinement and correction. We built our work upon SUSTech POINTS with an improved annotation propagation algorithm. In addition, we extend its functions to include auto-labeling using an MOT tracker, orientation adjustment, trajectory smoothing, etc.

Point Cloud Benchmark Datasets. Many point cloud benchmark datasets focus on autonomous driving applications in which the LiDAR is mounted on moving vehicles. For example, KITTI—a point cloud dataset released in 2013 and now a pioneering vision benchmark Geiger et al. (2013)—is widely used for evaluating perception tasks. Later, more autonomous driving datasets appeared, comprising more diverse scenes, larger sizes, and more fine-grained annotations such as Argoverse Chang et al. (2019), Nuscenes Caesar et al. (2020), and the Waymo Open DatasetSun et al. (2020).

Infrastructure-side point cloud benchmarks, on the other hand, are scarce. To our knowledge, the first infrastructure LiDAR dataset was released in 2014 and is referred to as the Ko-PER Intersection dataset Strigel et al. (2014). It deploys 14 SICK LD-MRS 8-layer research laser scanners to a four-way intersection in Aschaffenburg, Germany. Later in 2021, IPS300+ Wang et al. (2021a) released a high-density intersection dataset. It installs two 80-beam Robosense Ruby-Lite LiDAR scanners at the diagonal of a 4-way intersection. The two LIDAR cameras are calibrated and cover the entire intersection. However, only a total of 600-frame annotations are made available. Recently, LUMPI Busch et al. (2022) proposed a multi-perspective intersection dataset in Hanover, Germany. It deployed three cameras and 5 LiDARs to cover the intersection with dense point clouds. And a total of 90K point clouds have been released. However, their labels remain unavailable.

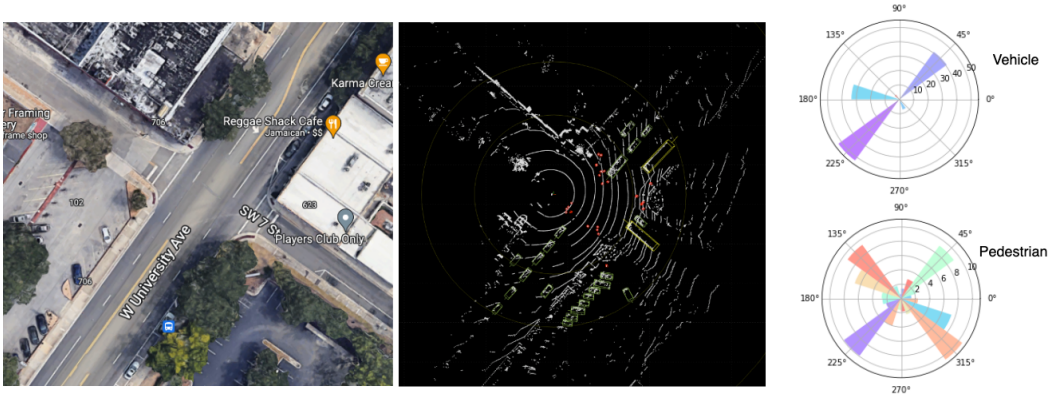
3 METHODOLOGY

This section introduces the collected dataset, detailed statistics and a qualitative comparison with other infrastructure LiDAR datasets. Then, we present an overview of the proposed semi-automated annotation scheme, followed by an explanation of the utilized deep-learning- based models.

3.1 THE FLORIDA DATASET

Data collection. We collected the dataset at a busy intersection—West University Avenue & Northwest 17th Street, Gainesville, FL—near the campus of the University of Florida. The LiDAR camera is mounted on a 5-meter post at the intersection. We used a Velodyne VLP-32C LiDAR with 32 channels, a 200-meter range, $+15^\circ$ to -25° vertical field of view (FOV), and 360° horizontal FOV. We manually selected 11 sequences, some of which included crowded pedestrians, abnormal behaviors, or near-misses. Such examples can be found in appendix A.1. Henceforth, we refer to our dataset as **FLORIDA**—**F**lorida **L**iDAR-based **O**bject **R**ecognition and **I**ntelligent **D**ata **A**nnotation.

Dataset statistics and characteristics. As shown in Table 2 and Figure 2 (c), we first summarize the statistics for all categories and display the orientation histogram of vehicles and pedestrians. The orientation histogram indicates that most vehicles move in a 45°/225° direction, corresponding to West University Avenue. The 165° direction comes from a parking lot where most vehicles park in parallel. For pedestrians, most of them cross the streets in the zebra-crossings, resulting in spikes in 45°, 225°, 135°, and 325° directions. We also compare our dataset with some of the popular infrastructure LiDAR datasets, as summarized in Table 1.



(a) Google map of the intersection (b) One annotated frame (c) Orientation histogram

Figure 2: We collect crowded pedestrian sequences from a LiDAR installed at a busy intersection near the campus of the University of Florida. (c) is the orientation histogram of vehicles and pedestrians, and the count numbers in the images are measured in thousands.

Table 2: The statistics of FLORIDA

Class	Vehicle	Pedestrian	Cyclist	Motorcycle	Bus	Truck
The total number of instances	143,941	80,220	999	17,397	4,170	2,640
The average number of instances per frame	21.81	12.15	0.15	2.64	0.63	0.40
The maximal number of instances per frame	38	34	2	7	3	3

3.2 Overview of the Semi-automated Annotation Algorithm

The overview of the semi-automated annotation algorithm is shown in Figure 1. For a new dataset, we begin by providing the initial annotation of an object and then use a SOT model Zheng et al. (2022) pre-trained on self-driving datasets to propagate annotation across subsequent frames. Of course, the auto-generated BBoxes might be imperfect; therefore, manual annotator refinement is necessary. The annotator can leverage batch mode adjustment and post-processing functions to reduce the refinement effort. Once a sequence is fully annotated, it is used to train SOT and MOT models. Both models are iteratively trained such that their tracking accuracy improves gradually with adding more annotated sequences to the training set. In contrast to SOT, MOT does not require first-frame annotation for each object and can be used to generate labels prior to any manual annotation. However, we found that MOT models pre-trained on self-driving datasets do not provide usable initial annotation, so only the SOT model is used to annotate the first few sequences. Once the MOT model has sufficient training data, it is able to provide the initial annotation, which, as demonstrated by experiments, significantly increases the annotation speed. Nevertheless, the MOT algorithm will still miss some objects or provide imprecise annotations. The annotator will then check each tracklet and make necessary adjustments. Additionally, the SOT can be utilized as a remedy for objects missed by MOT.

We implemented our annotation tool on top of SUSTechPOINTS, which includes box adjustment and batch-editing functions. We integrate the SOT tracker, post-processing, and trajectory-smoothing functions into the SUSTechPOINTS as drop-down menu options accessible via right-clicking a BBox. The returned BBox predictions or corrections are then converted into annotations and displayed on the front end. The MOT predictions are made prior to using the annotation tool. Specifically, we apply CenterPoint Yin et al. (2021) and SimpleTrack Pang et al. (2021) for multi-object detection and tracking respectively; we then convert the tracking results to SUSTechPOINT's annotation format. The annotation tool reads the MOT-predicted annotations and displays them.

Annotation Propagation by Single Object Tracking. Given a point cloud sequence and a BBox of an object in the first frame as the input of a 3D SOT tracker, we aim to locate the same object in a sequence of frames. Specifically, given a point cloud sequence $\{P^t\}_{t=1}^{T}$ of frame length T and the 3D BBox $B^1 \in \mathbb{R}^7$ of one object, parameterized by its location in 3D coordinates, height, length, width, and heading direction, at the first frame, a SOT tracker aims to find all 3D BBoxes of the object in subsequent frames denoted as $\{B^t\}_{t=2}^{T}$. To regress the position offset of the object between

Table 3: Comparative analysis of annotation speed and precision between SUSTechPOINTS and ours. Our annotation tool speeds up the annotation by more than $3\times$ while improving the annotation quality, especially for pedestrians.

	SUSTechPOINTS		Ours		Our Improvement	
	Car	Pedestrian	Car	Pedestrian	Car	Pedestrian
Avg time-trained (min)	80.5	119.5	21.0	31.0	$\times$3.8	$\times$3.9
Avg time-untrained (min)	152.0	185.0	34.5	47.5	$\times$4.4	$\times$3.9
Avg F-1 score-trained (%)	91.0	80.6	97.8	96.4	+6.8	+15.8
Avg F-1 score-untrained (%)	92.5	68.0	97.6	96.8	+5.1	+28.8

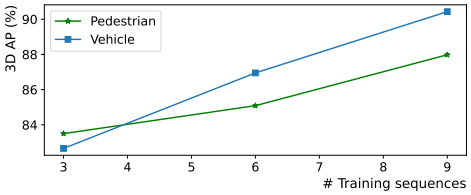


Figure 3: 3D AP varying the amount of training data. Training on 3 sequences already produces a decent result. The AP further improves when training on more sequences.

two consecutive frames, we resort to the state-of-the-art 3D SOT model—M^2-Track Zheng et al. (2022). The details can be found in the appendix A.2. In our setting, the annotator provides the initial BBox annotation, followed by the trained SOT tracker locating the object frame-by-frame. Because objects move continuously in 3D, their locations in two consecutive frames are close; therefore, the search area can be K meters around the object's last location. K is a hyper-parameter determined by the object's velocity and the frame rate. Following Zheng et al. (2022) and taking our static LiDAR setting into account, we empirically set K to 2 for vehicles and 0.5 for pedestrians.

We integrate the SOT model into SUSTechPOINTS Li et al. (2020) by replacing its original auto-labeling function with M^2-Track, resulting in a more robust function for handling occlusion and sparsity and producing better accuracy for deformable objects like pedestrians. We utilize the SOT model to a selected BBox to propagate to subsequent N frames, which returns N BBox predictions and displays them on the annotation tool. The parameter N is configurable. In addition, the annotator can switch to batch processing mode, where adjusting the keyframes will trigger interpolation for middle frames to speed up the annotation. Next, if the object is still visible after N frames, one can adjust the last-frame annotation and continue propagating the annotation to subsequent frames. To harmonize the SOT algorithm and interpolation, we set one annotation out of every ten as a keyframe such that the annotator can quickly refine the annotation by adjusting keyframes alone most of the time. Adjusting keyframes may be insufficient for turning vehicles, as the orientation change is non-linear. In this case, we can refine some annotations in non-keyframes. Once refined, a non-keyframe will change to a keyframe and accordingly trigger the interpolation.

Auto-annotation by Multi-Object Tracker. Given a point cloud sequence, the goal of MOT is to localize and identify all objects in the sequence. Formally, given point cloud sequence $\{P^i\}_{t=1}^{T}$, the MOT finds the BBoxes of all objects $\{\{B_j^t\}_{t=1}^{T}\}_{j=1}^{N^t}$, where N^t is the number of objects in frame P^t. Note that N^t varies over frames, as objects may enter or exit the scene at different times.

In our annotation scheme, an MOT model automatically generates tracklets for all objects in the scene. We apply CenterPoint Yin et al. (2021) for multi-object detection. Given the detection results in each frame, the next stage is associating the BBox of the same object across frames, producing tracklets of objects. To this end, we adopted SimpleTrack Pang et al. (2021), a top-performing multi-object tracking approach to generate object tracks. The details on SimpleTrack and additional post-process can be found in appendix A.3 and A.4. There are cases where the MOT model makes mistakes. For instance, we notice missing detection, incorrect detection, track ID switches, reversed orientation, etc. We further develop functions to assist annotators in quickly correcting errors. We leverage the SOT model for missing detection to propagate annotations for completion. For incorrect detection, annotators could delete all annotations for a given ID. To handle track ID switches, annotators could correct the ID where the switch happens and synchronize the change to the following frames. Lastly, they could correct the reversed orientation via a single one-click or batch correction in batch mode.

4 Experiments

In this section, we conduct several experiments to demonstrate the FLORIDA dataset's quality and the annotation scheme's usefulness. First, We evaluate the speed and accuracy of our developed annotation tool through annotator experiments. Next, we present baseline detection results with a study of the trade-off between annotation quantity and detection accuracy. Additionally, we illustrate the improvement in annotation speed as more data is annotated.

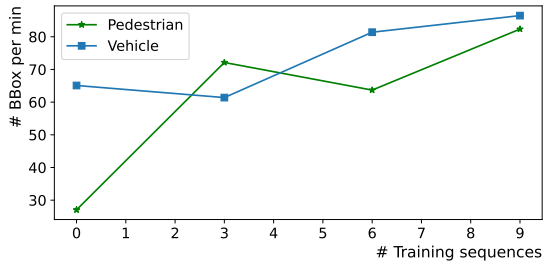


Figure 4: Annotation speed improvement as the MOT is trained on more sequences. # training sequences = 0 indicates using SOT annotation alone.

Table 4: 3D object detection result on FLORIDA dataset.

	Vehicle IoU = 0.7	Pedestrian IoU = 0.5	Motorcycle IoU = 0.5	Bus IoU = 0.7
3D AP (%)	90.66	87.44	82.32	91.99
BEV AP (%)	96.62	87.76	96.96	95.17

Annotator Experiments. We conduct a human annotation experiment to evaluate the efficiency and accuracy of our annotation tool. Specifically, we provide the average annotation times of four annotators, two trained and two untrained. We select a 200-frame LiDAR sequence with crowded vehicles and pedestrians and ask annotators to label it using two different annotation tools—SUSTechPOINTS and ours. Ground truth labels are annotated and double-checked by an experienced annotator. When annotating the ground truth, we verify the annotation by observing a longer sequence. The annotation efficiency is measured by the average annotation time, and the annotation accuracy is measured by the average F_1 score. We consider an annotation BBox to be accurate if the Intersection over Union (IoU) with a ground truth BBox exceeds a threshold. As the tightness of BBoxes differs between annotators, in the experiment, we set the IoU threshold at 0.3. Table 3 summarizes our tool's annotation efficiency and accuracy against SUSTechPOINTS. It shows that our annotation tool nearly quadruples the speed of annotation for both trained and untrained annotators. Meanwhile, our tool's annotation quality is also better, especially for pedestrians, due to that the MOT algorithm provides a template for the annotator to largely improves the recall for pedestrians, as detailed in appendix A.5.

3D Detection Results in FLORIDA. In Table 4, we show the 3D detection results on two 600-frame validation sequences. The validation sequences are collected on different days, non-overlapping with the training sequences. We use the 3D Average Precision (3D AP) and Bird's Eye View Average Precision (BEV AP) as evaluation metrics, as defined by the KITTI benchmark Geiger et al. (2013). We employ lower Intersection over Union (IoU) thresholds for smaller objects, such as Pedestrian and Motorcycle, and higher IoU thresholds for larger objects, such as Vehicle and Bus. The table excludes Truck and Cyclist due to insufficient instances. Moreover, we investigate the improvement of the detector's AP as more training data is gradually added. As shown in Figure 3, training on three 600-frame sequences already gives a reasonably good result, whereas the AP improvement from 3 to 6, and 6 to 9 are less significant. Therefore, given the detection accuracy requirement for different downstream tasks, one can vary the amount of annotation.

Annotation Time Reduction with Training on More Data. As shown in Figure 4, we record the annotation time for Vehicle and Pedestrian in the FLORIDA dataset to demonstrate the effectiveness of the human-in-the-loop concept. We train a new model for every three 600-frame sequences and calculate the average number of BBoxes per minute to measure annotation speed. As object density and moving patterns vary across different sequences, the resulting data points are fuzzy. However, we still observe a clear trend of increasing annotation speed. For Vehicle, annotation propagation with SOT and batch-mode interpolation already provide high annotation speed. For Pedestrian, MOT significantly increases speed. The MOT model trained on three sequences increases the annotation speed from 27.05 to 72.13 BBoxes/min.

5 Conclusion and Future Direction

In this paper, we have developed a semi-automated annotation tool that applies SOT and MOT models integrated with the human-in-the-loop schema for speeding up data annotation of challenging intersection LiDAR datasets. We verify its effectiveness via conducting human annotator experiments and reporting qualitative and quantitative results on object detection. Besides, we have introduced a fully-annotated infrastructure LiDAR perception dataset to facilitate research on infrastructure-based object perception. We provide an example in appendix A.3. In the future, we aim to enrich the dataset with video cameras and further reduce the annotation time.

REFERENCES

Hasan Asy'ari Arief, Mansur Arief, Guilin Zhang, Zuxin Liu, Manoj Bhat, Ulf Geir Indahl, Håvard Tveite, and Ding Zhao. SAnE: smart annotation and evaluation tools for point cloud data. *IEEE Access*, 8:131848–131858, 2020.

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. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Steffen Busch, Christian Koetsier, Jeldrik Axmann, and Claus Brenner. LUMPI: The Leibniz university multi-perspective intersection dataset. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1127–1134. IEEE, 2022.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. Nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631, 2020.

Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3D tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8748–8757, 2019.

Hsu-kuang Chiu, Jie Li, Rareş Ambruş, and Jeannette Bohg. Probabilistic 3D multi-modal, multi-object tracking for autonomous driving. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14227–14233. IEEE, 2021.

Peishan Cong, Xinge Zhu, Feng Qiao, Yiming Ren, Xidong Peng, Yuenan Hou, Lan Xu, Ruigang Yang, Dinesh Manocha, and Yuexin Ma. STCrowd: A multimodal dataset for pedestrian perception in crowded scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19608–19617, 2022.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3D Siamese tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1359–1368, 2019.

Le Hui, Lingpeng Wang, Mingmei Cheng, Jin Xie, and Jian Yang. 3D Siamese voxel-to-BEV tracker for sparse point clouds. *Advances in Neural Information Processing Systems*, 34:28714–28727, 2021.

Roy Jonker and Anton Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

E Li, Shuaijun Wang, Chengyang Li, Dachuan Li, Xiangbin Wu, and Qi Hao. SUSTech points: A portable 3D point cloud interactive annotation platform system. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1108–1115. IEEE, 2020.

Ziqi Pang, Zhichao Li, and Naiyan Wang. Simpletrack: Understanding and rethinking 3D multi-object tracking. *arXiv preprint arXiv:2111.09621*, 2021.

DSG Pollock et al. Smoothing with cubic splines, 1993.

Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep Hough voting for 3D object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9277–9286, 2019.

Charles R Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3D object detection from point cloud sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6134–6144, 2021.

Haozhe Qi, Chen Feng, Zhiguo Cao, Feng Zhao, and Yang Xiao. P2B: Point-to-box network for 3D object tracking in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6329–6338, 2020.

Elias Strigel, Daniel Meissner, Florian Seeliger, Benjamin Wilking, and Klaus Dietmayer. The Ko-PER intersection laserscanner and video dataset. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1900–1901. IEEE, 2014.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2446–2454, 2020.

Bernie Wang, Virginia Wu, Bichen Wu, and Kurt Keutzer. LATTE: accelerating LiDAR point cloud annotation via sensor fusion, one-click annotation, and tracking. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 265–272. IEEE, 2019.

Huanan Wang, Xinyu Zhang, Jun Li, Zhiwei Li, Lei Yang, Shuyue Pan, and Yongqiang Deng. Ips300+: a challenging multimodal dataset for intersection perception system. *arXiv preprint arXiv:2106.02781*, 2021a.

Zhoutao Wang, Qian Xie, Yu-Kun Lai, Jing Wu, Kun Long, and Jun Wang. MLVSNet: Multi-level voting Siamese network for 3D visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3101–3110, 2021b.

Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3D multi-object tracking: A baseline and new evaluation metrics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10359–10366. IEEE, 2020.

Hai Wu, Wenkai Han, Chenglu Wen, Xin Li, and Cheng Wang. 3D multi-object tracking in point clouds based on prediction confidence-guided data association. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3D object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11784–11793, 2021.

Chaoda Zheng, Xu Yan, Jiantao Gao, Weibing Zhao, Wei Zhang, Zhen Li, and Shuguang Cui. Box-aware feature enhancement for single object tracking on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13199–13208, 2021.

Chaoda Zheng, Xu Yan, Haiming Zhang, Baoyuan Wang, Shenghui Cheng, Shuguang Cui, and Zhen Li. Beyond 3D Siamese tracking: A motion-centric paradigm for 3D single object tracking in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8111–8120, 2022.

Changqing Zhou, Zhipeng Luo, Yueru Luo, Tianrui Liu, Liang Pan, Zhongang Cai, Haiyu Zhao, and Shijian Lu. PTTR: Relational 3D point cloud object tracking with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8531–8540, 2022.

Walter Zimmer, Akshay Rangesh, and Mohan Trivedi. 3D BAT: A semi-automatic, web-based 3D annotation toolbox for full-surround, multi-modal data streams. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1816–1821. IEEE, 2019.

A Appendix

A.1 Samples of the FLORIDA Dataset

As depicted in Figure 5, the FLORIDA dataset captures crowded pedestrians and vehicles and several abnormal behaviors.

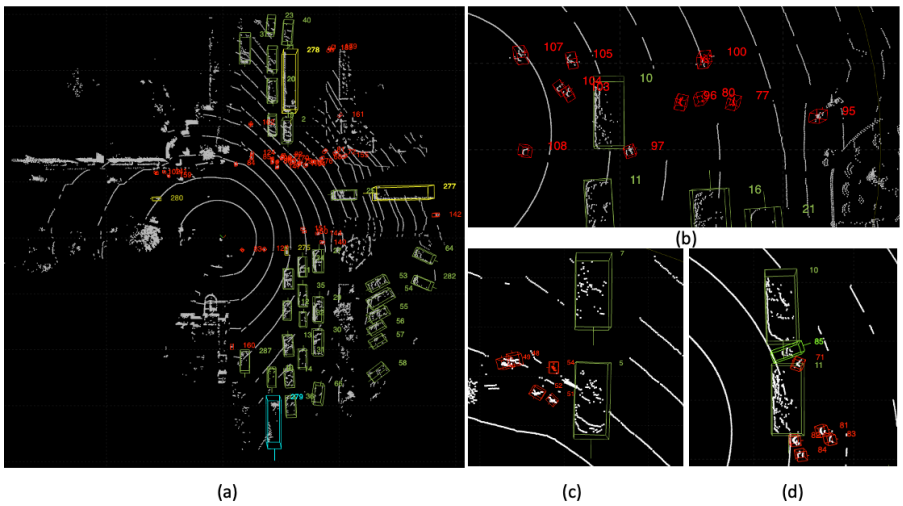


Figure 5: (a) A sample of a crowded scene. (b) A vehicle stopped at a pedestrian crosswalk. (c) People exiting a vehicle stopped at a red light. (d) A cyclist and a pedestrian passing through a small gap between two cars.

A.2 Details on the M^2-Track SOT Model

M^2-Track Zheng et al. (2022) proposes a two-stage motion-centric paradigm in which the motion transformation between the same objects in two frames is first regressed, followed by a refinement based on the merged point cloud in two frames. In detail, M^2-Track initially segments the target points in two frames using a trained semantic segmentation network. Then a motion vector $M = (\delta x, \delta y, \delta z, \delta\theta)$ is regressed by a motion estimation network, where $\delta x, \delta y, \delta z$ represent the location offsets, and $\delta\theta$ represents the heading direction angle offset. Adding the motion vector M to B^{t-1} gives us a coarsely predicted BBox $\hat{B}^t$. In the second stage, M^2-Track refines $\hat{B}^t$ by regressing a small relative offset and producing the final prediction B^t. Specifically, M^2-Track aggregates the previous frame point clouds P^{t-1} into the current point cloud P^t, compensating motion using the predicted M, resulting in a denser point cloud $\tilde{P}^t$. Another regression network is applied to $\tilde{P}^t$ to produce the refined BBox B^t.

A.3 Details on the SimpleTrack MOT Model

Following the "tracking-by-detection" paradigm, SimpleTrack unifies the 3D MOT methods into a general framework. The framework consists of four main components: (i) detection pre-processing, (ii) BBox association across frames, (iii) object motion modeling, and (iv) tracklet lifecycle management. Given multiple options in each component, we adopted those matching our dataset's characteristics. The pre-processing module mainly processes the raw detection predictions into a cleaner input to the tracker. We follow SimpleTrack to apply a stricter non-maximum suppression (NMS) to the raw detection predictions to preserve recall while improving precision. It effectively removes low-confidence detections that overlap with others while preserving low-confidence detections likely from sparse or occluded regions. For motion modeling, we adopted the Kalman Filter, which predicts the location of an object with increasing precision in the next frame. The Kalman Filter performs exceptionally well on infrastructure datasets because the LiDAR is stationary, resulting in longer tracklets and no abrupt motions. The predicted location from the motion model is then used as a proposal to associate with detections in the next frame. Next, for BBoxes association across frames, we view the problem as a bipartite matching problem and employ the well-known Hungarian algorithm Jonker & Volgenant (1987). As objects enter and exit the LiDAR's field of view at different times, the life cycle of tracklets needs to be carefully maintained. We adopt the "two-stage association" strategy of SimpleTrack. The detection score threshold is higher in the first stage than in the second. The first stage ensures tracking precision, while the second stage extends the life of tracklets in occluded or sparse regions, thereby reducing the number of ID switches.

We also post-process the produced tracks from SimpleTrack with heuristic rules to better aid annotation. First, we remove the tracklets that are too short because they are likely to be false positives. Second, we filter the tracklets whose speed is outside a reasonable range. For instance, the typical walking speed of a pedestrian is less than 2 meters per second. Therefore, predicted pedestrian tracklets with a higher average speed are more likely to be cyclists or motorcycles. Finally, because our dataset contains many parked cars, the bounding boxes of such tracklets vary slightly from frame to frame. Therefore, we average them to generate more accurate annotations.

A.4 Pre-processing and Post-processing Algorithms

A.4.1 Trajectory smoothing and orientation post-processing

When annotating pedestrians, we find it particularly challenging to determine their orientation from a single frame. For example, the point cloud on a pedestrian could be very sparse and incomplete. Often, the annotator has to examine the sequence surrounding the current frame to determine the orientation of a pedestrian based on movement. Therefore, we developed an orientation post-processing algorithm that imitates the annotator's behavior and significantly reduces the pedestrian annotation time. Specifically, after annotating a sequence, we first smooth the trajectory using a cubic smoothing spline algorithm Pollock et al. (1993) and then set the orientation at each timestamp as the pedestrian's moving direction. It is more tricky to set the orientation for stationary pedestrians. Therefore, we adopt a heuristic strategy: if the pedestrian starts moving later in the sequence, we set the orientation to match the direction of movement; otherwise, the orientation remains the same as the pedestrian's initial orientation.

A.4.2 Ground height estimation

For small objects (i.e., pedestrians and cyclists) that are too close or too far from the LiDAR's center, there are only a few points on each object, and it is ambiguous to determine the object's z value, i.e., height. The ground height information is helpful in such cases, as it allows us to better spot objects on the ground using SOT and MOT algorithms. To obtain the ground height, we manually segment the ground points using Point Cloud Labeler Behley et al. (2019). Next, the ground points are interpolated into grids using the `LinearNDInterpolator` from the python SciPy library. However, interpolation does not work well for distant regions with sparse data points. To cover these regions, we estimate a ground plane given all segmented ground points using the RANSAC algorithm Fischler & Bolles (1981). The interpolation captures subtle differences in ground height, such as the sidewalk being slightly higher than the road. Additionally, the ground plane captures the intersection's general elevation or the LiDAR's slight tilt angle. Note that the ground height of an intersection only needs to be estimated once.

A.5 A qualitative illustration of our tool's superiority over SUSTechPOINTS

Compared to SUSTechPOINTS, the pedestrian annotation quality of our tool is significantly higher, as shown in 3. The main reason is that the MOT algorithm provides a template for the annotator, which largely improves the recall for pedestrians. As shown in Figure 6, the annotator using SUSTechPOINTS does not recognize the pedestrians waiting to cross the street as human annotators recognize objects primarily based on motion, whereas the pedestrians in the blue box are stationary. On the other hand, our MOT algorithm enables the annotator to recognize and accurately annotate these pedestrians.

A.6 Application to traffic monitoring systems

We demonstrate the effectiveness of the FLORIDA dataset with our semi-automated annotation suite through a downstream use case. We integrate the predicted object trajectories into a web-based visual analytics system, where one can check all trajectories in a given time period, obtain count statistics for traffic participants, observe abnormal behaviors, etc. Compared with video sensors, LIDAR performs well regardless of the lighting conditions, thereby enhancing the safety of intersections.

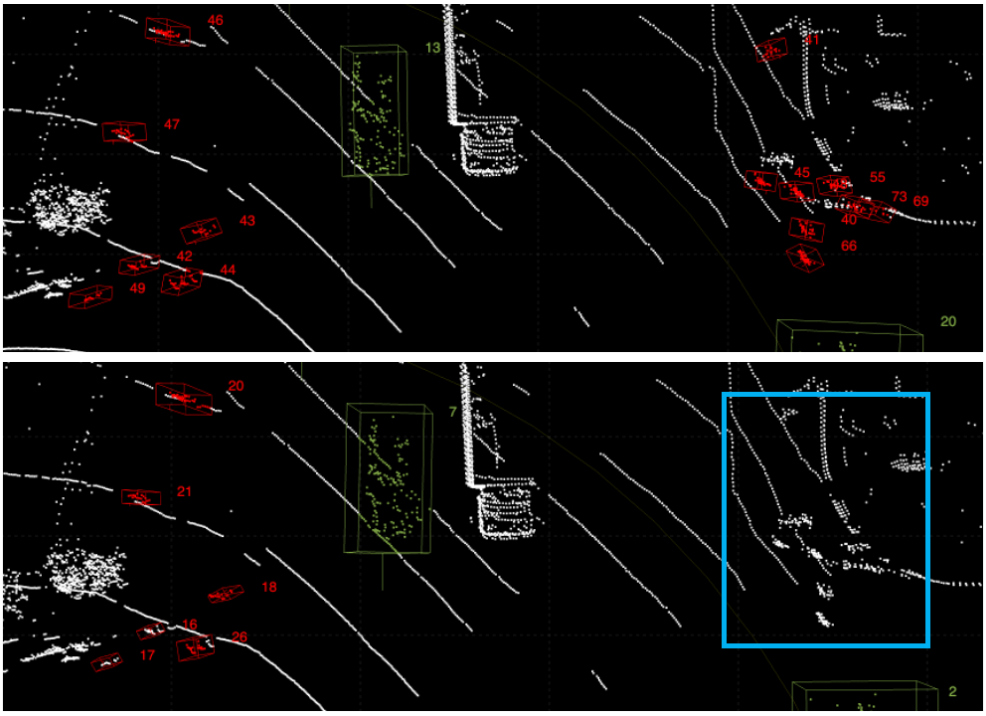


Figure 6: Example of annotations from an untrained annotator using our tool versus SUSTech-POINTS. The top one is annotated using our tool, while the bottom is annotated using SUSTech-POINTS. The pedestrians within the blue box are not recognized by the annotator.

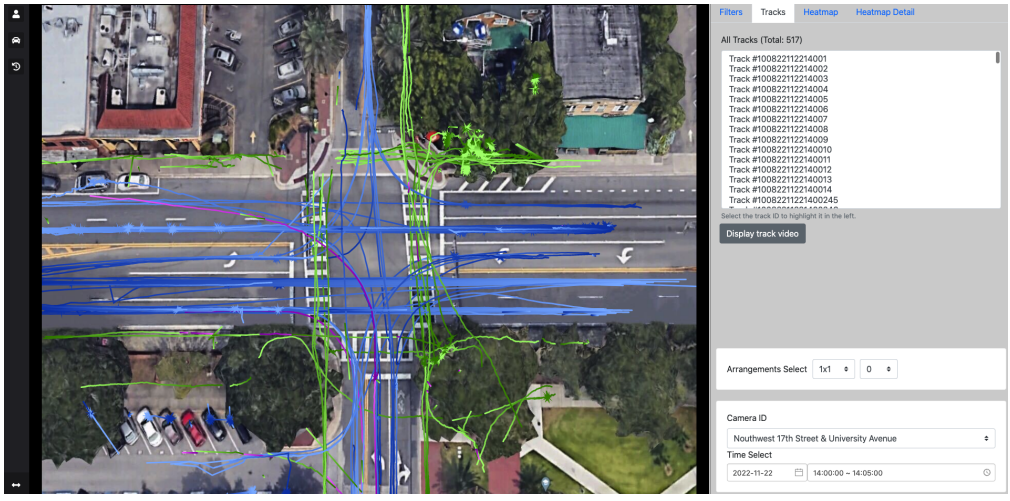


Figure 7: Application of our work on a web-based intersection traffic monitoring system. The trajectories of different types of traffic participants are shown in different colors. One can inspect all the trajectories and analytic statistics of a given time period.