nuScenes: A multimodal dataset for autonomous driving

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu,
Anush Krishnan, Yu Pan, Giancarlo Baldan, Oscar Beijbom
nuTonomy: an APTIV company
nuscenes@nutonomy.com

Abstract

Robust detection and tracking of objects is crucial for the deployment of autonomous vehicle technology. Image-based benchmark datasets have driven the development in computer vision tasks such as object detection, tracking and segmentation of agents in the environment. Most autonomous vehicles, however, carry a combination of cameras and range sensors such as lidar and radar. As machine learning based methods for detection and tracking become more prevalent, there is a need to train and evaluate such methods on datasets containing range sensor data along with images. In this work we present nuTonomy scenes (nuScenes), the first dataset to carry the full autonomous vehicle sensor suite: 6 cameras, 5 radars and 1 lidar, all with full 360 degree field of view. nuScenes comprises 1000 scenes, each 20s long and fully annotated with 3D bounding boxes for 23 classes and 8 attributes. It has 7x as many annotations and 100x as many images as the pioneering KITTI dataset. We also define a new metric for 3D detection which consolidates the multiple aspects of the detection task: classification, localization, size, orientation, velocity and attribute estimation. We provide careful dataset analysis as well as baseline performance for lidar and image based detection methods. Data, development kit, and more information are available at `www.nuscenes.org`.

1. Introduction

Autonomous driving technology has the potential to radically change the cityscape and save many human lives [52]. A crucial part of safe navigation is the detection and tracking of agents in the environment surrounding the vehicle. To achieve this, a modern self-driving vehicle deploys several sensors along with sophisticated detection and tracking algorithms. Such algorithms rely increasingly on machine learning, which drives the need for benchmark datasets for training and evaluation. While there is a plethora of image datasets for this purpose, there is a lack of large-scale multimodal datasets that cover 360°. We release the nuScenes dataset to address this gap.

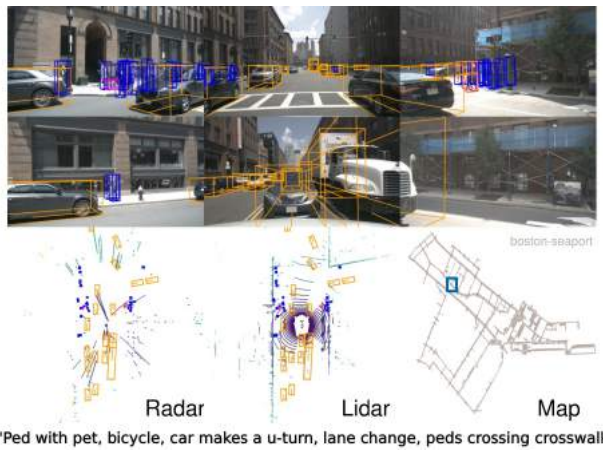

Figure 1. An example from the nuScenes dataset. We see 6 different camera views, lidar and radar data, as well as the human annotated semantic map. At the bottom we show the human written scene description.

Multimodal datasets are of particular importance as no single type of sensor is sufficient and the sensor types are complementary. Cameras allow accurate measurements of edges, color and lighting enabling classification and localization on the image plane. However, 3D localization from images is challenging [7, 6, 39, 54, 46, 43, 50]. Lidar pointclouds, on the other hand, contain less semantic information but highly accurate localization in 3D [35]. Furthermore the reflectance of lidar is an important feature [29, 35]. However, lidar data is sparse and the range is typically limited to 50-100m. Radar sensors achieve a range of 200-300m and measure the object velocity through the Doppler effect. However, the returns are even sparser than lidar and less precise in term of localization. While radar has been used for decades [1, 3], we are not aware of any autonomous driving datasets that provide radar data.

Since the three sensor types have different failure modes during difficult conditions, the joint treatment of sensor data is essential for agent detection and tracking. Liter-

Figure 2. Images from front camera collected from clear weather (col 1), nighttime (col 2), rain (col 3) and construction zones (col 4).

ature [32] even suggests that multimodal sensor configurations are not just complementary, but provide redundancy in the face of sabotage, failures, adverse conditions and blind spots. And while there are several works that have proposed fusion methods based on cameras and lidar [34, 8, 42, 36, 55, 51], PointPillars [35] has recently shown that a lidar only method currently performs on par, or even stronger than fusion based methods. This suggests that more work is required to combine the multimodal measurements in a principled manner.

In order to train such fusion-based methods, quality data annotations are required. Most datasets provide 2D semantic annotations as either boxes or masks (class or instance) [5, 12, 23, 57, 38]. Only a few datasets annotate objects using 3D boxes [22, 30, 41], and they do not provide the full sensor suite. To the best of our knowledge, none of these 3D datasets provide annotation of object attributes, such as pedestrian pose or vehicle state.

Existing AV datasets and vehicles are highly specialized for particular operational design domains. As suggested in [25], more research is required on how to generalize to "complex, cluttered and unseen environments". Therefore, we need to study how fusion-based methods will generalize to different countries, different lighting (daytime vs nighttime), driving directions, signage, road markings, vegetation, precipitation and previously unseen object types.

Contextual knowledge using semantic maps is also an important prior for scene understanding [56, 2, 24]. For example, one would expect to find cars on the road, but not on the sidewalk or inside buildings. While semantic categories can be inferred at inference time, manual labeling of the semantic labels of *static* background is typically more accurate. With the notable exception of [49], the majority of the AV datasets do not provide semantic maps.

1.1. Related datasets

The last decade has seen the release of several driving datasets which have played a huge role in scene-understanding research for Autonomous Vehicles (AV). Most of these datasets have focused on 2D annotations (bounding-boxes, segmentation polygons) for RGB camera images. The CamVid [5] dataset released four HD videos with semantic segmentation annotations for 701 images. Cityscapes [12] released stereo video sequences captured from 50 different cities with high quality pixel-level annotations for 5k images. Mapillary Vistas [23], BDD100k [57] and Apolloscape [30] released even larger datasets containing segmentation masks for 25k, 100k, and 144k images respectively. Vistas and BDD100k also contain images captured during different weather and illumination settings. Other datasets [13, 18, 53, 17, 59, 15, 40] focus exclusively on pedestrian annotations on images.

The ease of capturing and annotating RGB images have made the release of these large-scale image-only datasets possible. On the other hand, multimodal datasets, which are typically comprised of images, range sensor (lidars, radars) data and GPS/IMU data, are expensive to collect and annotate due to the difficulties of integrating, synchronizing, and calibrating multiple sensors. KITTI [22] was the pioneering multimodal dataset providing dense pointclouds from a lidar sensor as well as front-facing stereo images and GPS/IMU data. It provides 200k 3D boxes over 22 scenes which helped advance the state-of-the-art in 3D object detection. The recent H3D dataset [41] includes 160 crowded scenes with a total of 1.1M 3D boxes annotated over 27k frames. The objects are annotated in the full 360° view, as opposed to KITTI where an object is only annotated if it is present in the frontal view. They provide data from lidar, 3 cameras and GPS/IMU. However, the 3 cameras are all front facing which means that the vision sensors only provide 180° coverage. Further, both KITTI and H3D do not provide any radar or nighttime data. The KAIST multispectral dataset [10] is a multimodal dataset that consists of

Dataset	Year	# scenes	Size (hr)	# rgb imgs	# pc lidar	# pc radar	# ann. frames	# 3D boxes	Night	Rain	Snow	Locations
CamVid [5]	2008	4	0.4	18k	0	0	700	0	No	No	No	Cambridge
Cityscapes [12]	2016	-	-	25k	0	0	25k	0	No	No	No	50 cities
Vistas [23]	2017	-	-	25k	0	0	25k	0	*Yes*	*Yes*	*Yes*	6 continents
BDD100k [57]	2017	*100k*	*1k*	*100M*	0	0	100k	0	*Yes*	*Yes*	*Yes*	NY, SF
M. obj. det. [48]	2017	-	-	7.5k	0	0	7.5k	0	*Yes*	No	No	Tokyo
M. sem. seg. [26]	2017	-	-	1.6k	0	0	1.6k	0	*Yes*	No	No	-
ApolloScape [30]	2018	-	100	144k	0*	0	*144k*	*70k*	*Yes*	No	No	China†
KITTI [22]	2012	22	1.5	15k	15k	0	15k	200k	No	No	No	Karlsruhe
AS lidar [37]	2018	-	2	0	20k	0	20k	475k	-	-	-	-
KAIST [10]	2018	-	-	8.9k	8.9k	0	8.9k	0	**Yes**	No	No	Seoul
H3D [41]	2019	160	0.77	83k	27k	0	27k	1.1M	No	No	No	SF
nuScenes	2019	**1k**	**5.5**	**1.4M**	**400k**	**1.3M**	**40k**	**1.4M**	**Yes**	**Yes**	No	Boston, SG

Table 1. AV datasets comparison. SG: Singapore, NY: New York, SF: San Francisco, M: Multispectral, AS: ApolloScape. The table is split into two parts based on whether range data was provided (lower-half) or not (upper-half). Only datasets which provide annotations for at least *car*, *pedestrian* and *bicycle* are included in this comparison. "-" indicates that no information is provided. *[30] provides static depth maps. Moving objects in the scene (e.g. cars) are not present in those depth maps. †collected in "four regions" in China.

RGB/thermal camera, RGB stereo, 3D lidar and GPS/IMU. It provides nighttime data, but the size of the dataset is limited and annotations are in 2D. Other notable multimodal datasets include [9] providing driving behavior labels, [31] providing place categorization labels and [4, 38] providing raw data without semantic labels.

An alternative to collecting real-world multimodal driving data is by generating synthetic data via simulation. CARLA [16], SYNTHIA [47], and Virtual KITTI [21] simulate virtual cities using game engines. Playing for Benchmarks [45] retrieves renderings and annotations from GTA without access to their source code. These have the advantage of simulating arbitrarily situations and avoiding the cost of human annotation. However, for the foreseeable future the generated images are not photo-realistic and can therefore not replace real-world datasets.

1.2. Contributions

From the complexities of the multimodal 3D detection challenge, and the limitations of current AV datasets, a large-scale multimodal dataset with 360° coverage across all vision and range sensors collected from diverse situations alongside map information would boost AV scene-understanding research further. nuScenes does just that, and it is the main contribution of this work.

nuScenes represents a large leap forward in terms of data volumes and complexities (Table 1), and is the first dataset to provide 360° sensor coverage from the *entire sensor suite*. It is also the first AV dataset to include *radar data* and the first captured using an AV *approved for public roads*. It is further the first multimodal dataset that contains data from *nighttime* and *rainy* conditions, and that carries annotation of object *attributes* in addition to class and location. nuScenes thus enables research on 3D object detection, 3D tracking, behavior modeling and prediction and trajectory estimation.

Our second contribution is a new detection metric that summarizes all aspects of 3D object detection into a single metric. We also train 3D object detectors as a baseline reference and to help guide future avenues of research. These baselines include a novel approach of using multiple lidar sweeps to enhance object detection. All data, code, and information is made available at `www.nuscenes.org`.

2. The nuScenes dataset

Here we describe how we plan drives, setup our vehicles, select interesting scenes, annotate the dataset and protect the privacy of third parties.

Drive planning. We drive in Boston (Seaport and South Boston) and Singapore (One North, Holland Village and Queenstown), two cities that are known for their dense traffic and highly challenging driving situations. We emphasize the diversity across locations in terms of vegetation, buildings, vehicles, road markings and right versus left-hand traffic. From a large body of training data we manually select 84 logs with 15h of driving data (242km travelled at an average of 16km/h). Driving routes are carefully chosen to capture a diverse set of locations (urban, residential, nature and industrial), times (day and night) and weather conditions (sun, rain and clouds).

Car setup. We use two Renault Zoe supermini electric cars with an identical sensor layout to drive in Boston and Singapore. See Figure 3 for sensor placements and Table 2 for sensor details. Front and side cameras have a 70° FOV and are offset by 55°. The rear camera has a FOV of 110°.

Sensor	Details
6x Camera	RGB, 12Hz capture frequency, $1/1.8$" CMOS sensor, 1600×900 resolution, auto exposure, JPEG compressed
1x Lidar	Spinning, 32 beams, 20Hz capture frequency, 360° horizontal FOV, -30° to 10° vertical FOV, $\leq$ $70m$ range, $\pm$2cm accuracy, up to $1.4M$ points per second.
5x Radar	$\leq$ $250m$ range, 77GHz, FMCW, 13Hz capture frequency, ± 0.1km/h vel. accuracy

Table 2. Sensor data in nuScenes.

Sensor calibration. To achieve a high quality multi-sensor dataset careful calibration or sensor intrinsic and extrinsic parameters is required. We express extrinsic coordinates of each sensor to be relative to the *ego frame*, i.e. the midpoint of the rear vehicle axle, using tools like laser liner and calibration target boards.

Sensor synchronization. In order to achieve good cross-modality data alignment between the lidar and the cameras, the exposure of a camera is triggered when the top lidar sweeps across the center of the camera's FOV. The timestamp of the image is the exposure trigger time; and the timestamp of the lidar scan is the time when the full rotation of the current lidar frame is achieved. Given that the camera's exposure time is nearly instantaneous, this method generally yields good data alignment [1].

Localization and map. Most existing datasets provide the vehicle location based on GPS+IMU [22, 30]. As we operate in dense urban areas, we find that GPS signals are not always reliable. To accurately localize our vehicle, we create a detailed prior map of lidar points in an offline step. On the car we use a Monte Carlo Localization scheme from lidar and odometry information [11]. This method is very robust and we achieve localization errors of $\leq 10cm$. We also provide highly accurate semantic maps of the relevant areas with a resolution of 10px/m. These human-annotated maps provide information on roads and sidewalks. We encourage the use of localization and semantic maps as strong priors for object detection, tracking and other tasks (e.g. pedestrians are typically found on sidewalks or crosswalks).

Scene selection. After collecting the raw sensor data, we manually select 1000 *interesting* scenes of $20s$ duration each. Interesting scenes include scenes with high traffic density (e.g. intersections, construction sites), rare classes (e.g. ambulances, animals), potentially dangerous traffic situations (e.g. jaywalkers, incorrect behavior), maneuvers (e.g. lane change, turning, stopping) and situations that may be difficult for an AV. We also select some scenes to encourage diversity in terms of spatial coverage, different scene types, as well as different weather and lighting conditions. Expert annotators write textual descriptions or *captions* for each scene (e.g.: "Wait at intersection, peds on sidewalk, bicycle crossing, jaywalker, turn right, parked cars, rain").

[1]The cameras run at 12Hz while the lidar runs at 20Hz. The 12 camera exposures are spread as evenly as possible across the 20 lidar scans, so not all lidar scans have a corresponding camera frame.

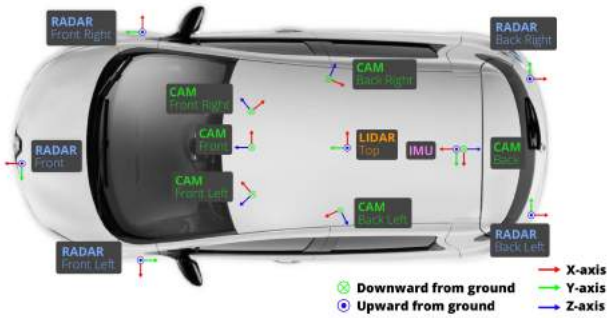

Figure 3. Sensor setup for our data collection platform.

Data annotation. Having selected the scenes, we sample keyframes (image, lidar, radar) at 2Hz. We annotate each of the 23 object classes in every keyframe in the form of cuboids modeled as x, y, z, width, length, height and yaw angle. We annotate objects continuously throughout each 20s scene if they are covered by at least one lidar or radar point. This provides temporal context so we can exploit multiple lidar sweep configuration in pointclouds and velocity/trajectory estimation. Using expert annotators and multiple validation steps, we achieve highly accurate annotations. All objects in the nuScenes dataset come with a semantic category, a 3D bounding box, and attributes (visibility, activity and pose) for each frame they occur in.

Privacy protection. It is our priority to protect the privacy of third parties. As manual labeling of faces and license plates is prohibitively expensive for 1.4M images, we use state-of-the-art object detection techniques. Specifically for plate detection, we use Faster R-CNN [44] with ResNet-101 backbone [28] trained on Cityscapes [12][2]. For face detection, we use [58][3]. We set the classification threshold to achieve an extremely high recall (similar to [20]). To increase the precision, we remove predictions that do not overlap with the reprojections of the known *pedestrian* and *vehicle* boxes in the image. Eventually we use the predicted boxes to blur faces and license plates in the images.

Data format. Contrary to most existing datasets [22, 41, 30], we store the annotations and metadata (e.g. localization, timestamps, calibration data) in a relational database which avoids redundancy and allow for efficient access. The nuScenes devkit, taxonomy and annotator instructions can be found in the devkit[4].

[2]https://github.com/bourdakos1/Custom-Object-Detection
[3]https://github.com/TropComplique/mtcnn-pytorch
[4]`https://github.com/nutonomy/nuscenes-devkit`.

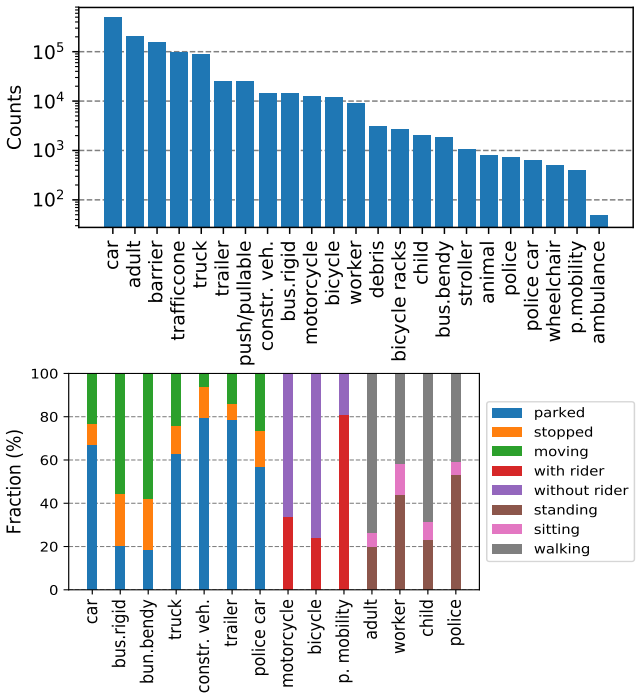

Figure 4. Top: Number of annotations per category. Bottom: Attributes distribution for selected categories. Cars and adults are the most frequent categories in our dataset while ambulance is the least frequent. The attribute plot also shows some expected patterns: construction vehicles are rarely moving, pedestrians are rarely sitting while buses are commonly moving.

3. Analysis

We analyze statistics of the annotations in nuScenes. Our dataset has 23 categories including different vehicles, types of pedestrians, mobility devices and other objects as seen in Figure 4. Statistics on geometry and frequencies of different classes are shown in Figure 5. Per keyframe there are 7 pedestrians and 20 vehicles on average. Moreover, 40k keyframes were taken from four different scene locations (Boston: 55%, SG-OneNorth: 21.5%, SG-Queenstown: 13.5%, SG-HollandVillage: 10%) with various weather and lighting conditions (rain: 19.4%, night: 11.6%). Figure 7 shows the map locations with spatial coverage across all scenes where the most coverage comes from intersections.

Figure 6 shows that *car* annotations are seen at varying distances and as far as 80m from the ego-vehicle. Box orientation is also varying, with the most number in vertical and horizontal angles for cars as expected due to parked cars and cars in the same lane.

Lidar and radar points statistics inside each box annotation are shown in Figure 8. Our annotations have up to 100 lidar points even at a radial distance of 80m and at most 12k lidar points at 3m. At the same time they contain up to 40 radar returns at 10m and 10 at 50m. The radar range far exceeds the lidar range at up to 200m.

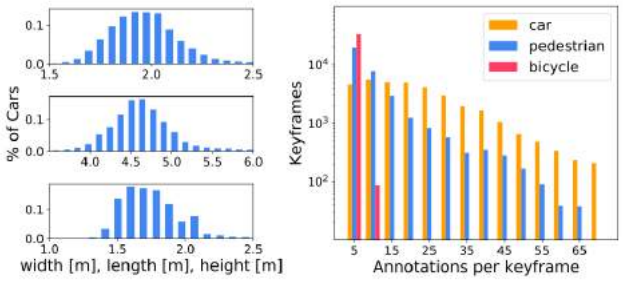

Figure 5. Left: Bounding box size distributions for *car*. Right: Category count in each keyframe for *car*, *pedestrian*, and *bicycle*.

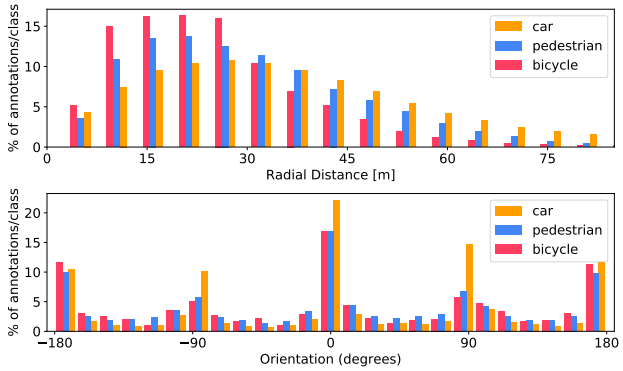

Figure 6. Top: radial distance of objects from the ego vehicle. Bottom: orientation of boxes in box coordinate frame.

To analyze the quality of our localization data, we compute the merged point cloud of an entire scene by registering approximately 800 point clouds in global coordinates. We remove points corresponding to the ego vehicle and assign to each point the mean color value of the closest camera pixel that the point is reprojected to. Scene reconstruction can be seen in Figure 9 which demonstrate accurate synchronization and localization.

4. Detection Task

This section outlines the metrics for the nuScenes detection task. In the future we may add other tasks and metrics.

The nuScenes detection task requires detecting 10 object classes with full 3D bounding boxes, attributes, and velocities. The 10 classes are a subset of all 23 classes annotated in nuScenes (details in the devkit).

We design metrics for each of these aspects and a schema for consolidation into a scalar score indicating method performance, the nuScenes detection score (NDS):

$$\text{NDS} = \frac{1}{10}[5\ \text{mAP} + \sum_{\text{mTP}\in\mathbb{TP}} (1 - \min(1, \text{mTP}))]. \quad (1)$$

Here mAP is mean Average Precision (2), and $\mathbb{TP}$ the set of the five mean True Positive metrics (3). Half of NDS is thus based on the detection performance while the other half quantifies the quality of the detections in terms of box location, size, orientation, attributes, and velocity.

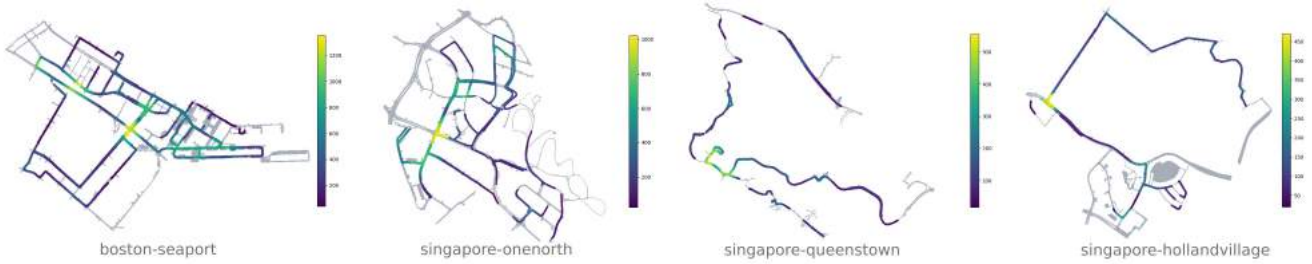

Figure 7. Spatial data coverage for all four locations. Colors indicate the number of keyframes with ego vehicle poses within a 100m radius across all scenes.

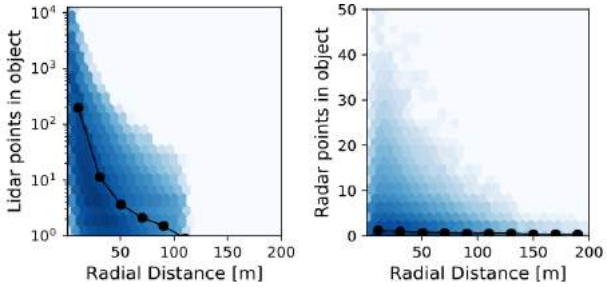

Figure 8. Hexbin log-scaled density plots of the number of lidar and radar points inside a box annotation. The black line represents the mean number of points for a given distance wrt the ego-vehicle.

Data Setup. Since the nuScenes dataset contains continuous 20s scenes, one could use data from the beginning of the scene to time t to determine the object locations at t. Indeed, this is what a production system would, and should do. However, for the purpose of this benchmark, and in order to separate out the performance of a *tracker* from a *detector*, we define the detection task to only operate on sensor data between $[t-0.5, t]$ seconds. Subsequent tracking tasks will allow data between $[0, t]$.

Average Precision metric. We use the Average Precision (AP) metric [22, 19], but define a match by thresholding the 2D center distance d on the ground plane instead of intersection over union. This is done in order to decouple detection from object size and orientation but also because small objects, like pedestrians, have such small footprints that a small translation error results in a zero intersection over union, which makes it hard to compare the performance of vision-only methods which tend to have large location errors [46].

We then calculate AP as the normalized area under the precision recall curve for recall and precision over 10%. Operating points where recall or precision is less than 10% are removed for two reasons: First, the measurement can be noisy in these regions, in particular for low recalls. Second, such extreme operating points would be highly unsuitable for deployment on public roads. If no operating point in this region is achieved, the AP for that class is set to zero. We finally average over matching thresholds

Figure 9. Sample scene reconstruction given lidar points and camera images. We project the lidar points in an image plane with colors assigned based on the pixel color from camera data.

of $\mathbb{D} = \{0.5, 1, 2, 4\}$ meters and the set of classes $\mathbb{C}$:

$$\text{mAP} = \frac{1}{|\mathbb{C}||\mathbb{D}|} \sum_{c \in \mathbb{C}} \sum_{d \in \mathbb{D}} \text{AP}_{c,d} \tag{2}$$

True Positive metrics. In addition to AP, we measure a set of *true positive metrics* (TP metrics) for each prediction that was matched with a ground truth box. All TP metrics are calculated using $d = 2$m center distance during matching, and they are all designed to be positive scalars. Matching and scoring happen independently per class and each metric is the average of the cumulative mean at each achieved recall levels above 10%. If 10% recall is not achieved for a particular class, all TP errors for that class is set to 1. The following TP errors are defined:

Average Translation Error (ATE) is the Euclidean center distance in 2D (units in $meters$). Average Scale Error (ASE) is the 3D IOU after aligning orientation and translation $(1 - IOU)$. Average Orientation Error (AOE) is the smallest yaw angle difference between prediction and ground-truth ($radians$). All angles are measured on a full 360° period except for barriers where they are measured on a 180° period. Average Velocity Error (AVE) is the absolute velocity error as the L2 norm of the velocity differences in 2D (m/s). Average Attribute Error (AAE) is defined as 1 minus attribute classification accuracy $(1 - acc)$. Finally, the mTP is calculated as:

$$\text{mTP} = \frac{1}{|\mathbb{C}|} \sum_{c \in \mathbb{C}} \text{TP}_c \tag{3}$$

Here we omit a few measurements that are not well defined for each class: AVE for cones and barriers since they are stationary; AOE of cones since they do not have a well defined

# Lidar Sweeps	Pretraining	NDS (%)
1	KITTI	31.8
5	KITTI	42.9
10	KITTI	44.8
10	ImageNet	**44.9**
10	None	44.2

Table 3. Lidar only baselines performance in terms of NDS (1). These experiments examine the effect of pretraining and number of sweeps on detection performance.

Class	**AP**	**ATE**	**ASE**	**AOE**	**AVE**	**AAE**
Car	75.9	0.27	0.17	0.19	0.27	0.47
Pedestrian	63.7	0.28	0.28	0.36	0.26	0.15
Bus	44.7	0.54	0.21	0.14	0.45	0.47
Barrier	42.8	0.69	0.32	0.10	N/A	N/A
Traffic Cone	32.9	0.45	0.41	N/A	N/A	N/A
Truck	31.9	0.53	0.24	0.14	0.21	0.41
Trailer	23.5	0.93	0.23	0.37	0.19	0.30
Motorcycle	22.8	0.38	0.30	0.64	0.48	0.60
Constr. Veh.	6.3	0.95	0.50	1.26	0.11	0.36
Bicycle	2.1	0.40	0.28	0.87	0.33	0.51
Mean	29.4	0.54	0.29	0.45	0.29	0.41

Table 4. Detailed detection performance for best lidar network. AP: average precision (%), ATE: average translation error (m), ASE: average scale error (1-IOU), AOE: average orientation error (rad), AVE: average velocity error (m/s), AAE: average attribute error ($1 - acc.$), N/A: not applicable (Sec 4). nuScenes Detection Score (NDS) = 44.9%

orientation; and AAE for cones and barriers since there are no attributes defined on these classes (Figure 4).

Note that since mAVE, mAOE and mATE can be larger than 1, we bound each metric between 0 and 1 when calculating NDS (1).

5. Experiments

In this section we present object detection experiments on the nuScenes dataset to serve as reference baselines and suggest avenues for future research.

5.1. Lidar baseline

To demonstrate the performance of a leading algorithm on nuScenes, we train a lidar only 3D object detector, PointPillars [35]. We take advantage of temporal data available in nuScenes by accumulating lidar sweeps for a richer point-cloud input to the point pillar encoder. A single PointPillars network was trained to predict 3D boxes for all classes. The previously published PointPillars network was modified to also learn velocities as an additional regression target for each 3D box. In these experiments we set the box attributes to the most common attribute for each class in the train data, and future work will explore jointly learning attributes with the other outputs.

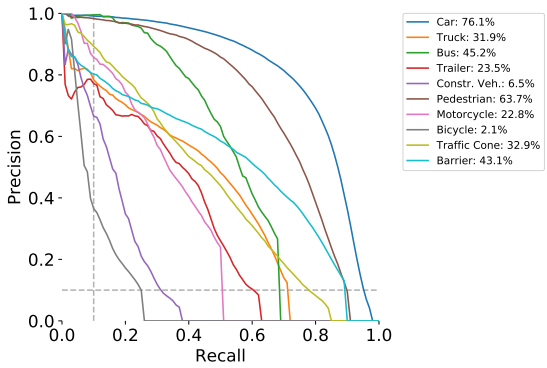

Figure 10. Precision vs recall for best lidar baseline using a 2m threshold for matching.

We investigate PointPillars performance by varying two important hyperparameters: the number of lidar sweeps and the type of pre-training.

Number of lidar sweeps. According to our evaluation protocol (Section 4), one is only allowed to use $0.5s$ of previous data to make a detection decision. This corresponds to 10 previous lidar sweeps since the lidar is sampled at 20 Hz. Accumulation is implemented by moving all point clouds to the coordinate system of the keyframe and appending a scalar time-stamp to each point indicating the time delta in seconds from the keyframe. The PointPillars encoder includes the time delta as an extra decoration for the lidar points. Aside from the advantage of a richer point cloud input for detection, this also provides inherent temporal information in a single input which helps the network in localization and enables velocity prediction. We experiment with using 1, 5, and 10 lidar sweeps. [5]

Backbone pretraining. We examine whether features obtained from other domains or datasets generalize to nuScenes. No pretraining means weights are initialized randomly using a uniform distribution as in [27]. ImageNet [14] pretraining [33] uses a backbone that was first trained to accurately classify images. KITTI [22] pretraining uses a backbone that was trained on the lidar point-clouds to predict 3D boxes.

Analysis. As shown in Table 3, increasing the number of lidar sweeps leads to better detection performance although the performance saturates with increasing number of sweeps. The increased point density provided by extra sweeps leads to higher mean average precision of 21.9%, 27.7%, and 28.8% for 1, 5, and 10 sweeps respectively. Additionally, the temporal information provides context for learning velocities with an AVE of 1.21 m/s, 0.34 m/s, and 0.3 m/s respectively.

[5] 1, 5, and 10 lidar sweeps is, in practice, only 1, 4.9, and 9.8 sweeps on average: (1) the first keyframe of each scene has no previous sweeps and (2) limiting sweeps to the past 0.5s may discard the 10th sweep.

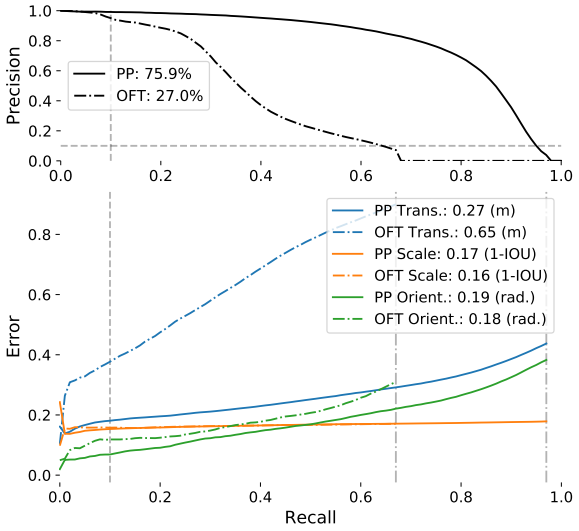

Figure 11. PointPillars (PP) [35] vs OFT [46] performance on *cars*. Top: precision vs recall using a 2m threshold for matching. Bottom: translation, orientation, and scale true positive metrics. Attributes and velocities were not learned and hence are omitted.

Interestingly, while the KITTI pretrained network did converge faster, the final performance of the network only marginally varied between different pretrainings. Since the ImageNet pretraining was the best, we will use that to examine performance in more detail and all analysis refers to this network. The per class performance on the nuScenes detection metrics is shown in Table 4 and Figure 10. The network performed best overall on cars and pedestrians which are the two most common categories. The worst performing categories were bicycles and construction vehicles, two of the rarest categories that also present additional challenges. However, the network achieved a bicycle mAP of approximately 30% when filtering for only predictions and ground truth on the semantic map. Bicycles that are not on the semantic map are especially difficult to detect because they are usually parked and often occluded or only visible from the front or back. Construction vehicles pose a unique challenge due to their high variation in size and shape. While the translational error is similar for cars and pedestrians, the orientation errors for pedestrians (21 deg) is higher than that of cars (11 deg). This smaller orientation error for cars is expected since cars have a greater distinction between their front and side profile relative to pedestrians. The vehicle velocity estimates are promising (e.g. 0.27 m/s AVE for the *car* class) considering the typical speed of a vehicle in the city would be 10 to 15 m/s.

5.2. Image baseline

To examine image-only 3D object detection, we adapt and train a leading algorithm, Orthographic Feature Transform (OFT) [46] on nuScenes. A single OFT network was used for all classes. We modified the original OFT implementation to use a SSD detection head and confirmed that this architecture matched published results on KITTI. The network takes in a single camera image and the full 360° predictions were obtained by using non-maximal suppression (NMS) to combine together the independent predictions from all 6 cameras. In this experiment we set the box velocity to zero and attributes to the most common attribute for each class in the train data.

Analysis. As shown in Figure 11, the OFT baseline achieved promising performance on *car* category and future work will be required to adapt OFT to the complexities of nuScenes to achieve higher performance on all categories. Comparing OFT and PointPillars in Figure 11 shows that PointPillars achieved a significantly higher average precision and max recall. However, OFT and PointPillars achieved a similar scale error over all recalls, demonstrating that object scale is equally well inferred from images or lidar. As expected, PointPillars has lower localization error than OFT since lidar points provide range information while OFT has to learn to associate range information with image only features. When averaged over all recalls, PointPillars and OFT had similar orientation error, but as shown in Figure 11, PointPillars achieved lower orientation errors when compared over the same recall. This shows that it is either important to compare the true positives over the same recall or to consider true positive metrics and average precision in one metric as in NDS (1).

5.3. Discussion

The two baselines demonstrate that while lidar only or image only detectors are both able to achieve promising detection results on cars, lidar only networks currently provide superior performance. Each sensor modality provides complimentary features for training 3D object detection and we encourage research on a fusion network that uses all sensor data (image, lidar, radar) as well as exploits prior information from semantic maps to achieve the best performance.

6. Conclusion

In this paper we present the nuScenes dataset, metrics, and baseline results. This is the only dataset collected from an autonomous vehicle on public roads and the only dataset to contain the full 360° sensor suite (lidar, images, and radar). nuScenes has the largest collection of 3D box annotations of any public dataset. To spur research on 3D object detection for autonomous vehicles, we introduce a new detection metric that balances all aspects of detection performance. We demonstrate novel adaptations of leading lidar only and image only 3D object detectors on nuScenes. We hope this dataset will help accelerate research and development of autonomous vehicle technology.

Acknowledgements. The nuScenes dataset was annotated by Scale.ai and we thank Alexandr Wang and Dave Morse for their support. We also thank Sun Li and Karen Ngo at nuTonomy for data inspection and quality control, and Bassam Helou for OFT baseline results. We thank Thomas Roddick for useful discussions about OFT.

References

[1] G. Alessandretti, A. Broggi, and P. Cerri. Vehicle and guard rail detection using radar and vision data fusion. *IEEE Transactions on Intelligent Transportation Systems*, 2007.

[2] D. Barnes, W. Maddern, and I. Posner. Exploiting 3d semantic scene priors for online traffic light interpretation. In *IVS*, 2015.

[3] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner. Three decades of driver assistance systems: Review and future perspectives. *ITSM*, 2014.

[4] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueas, and J. González-Jiménez. The Málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. *IJRR*, 2014.

[5] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008.

[6] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals for accurate object class detection. In *NIPS*, 2015.

[7] X. Chen, L. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. In *CVPR*, 2016.

[8] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, 2017.

[9] Y. Chen, J. Wang, J. Li, C. Lu, Z. Luo, H. Xue, and C. Wang. Lidar-video driving dataset: Learning driving policies effectively. In *CVPR*, 2018.

[10] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon. Kaist multi-spectral day/night data set for autonomous and assisted driving. *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[11] Z. J. Chong, B. Qin, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus. Synthetic 2d lidar for precise vehicle localization in 3d urban environment. In *ICRA*, 2013.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[15] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 2012.

[16] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *ACRL*, 2017.

[17] M. Enzweiler and D. M. Gavrila. Monocular pedestrian detection: Survey and experiments. *PAMI*, 2009.

[18] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *CVPR*, 2008.

[19] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 2010.

[20] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent. Large-scale privacy protection in google street view. In *ICCV*, 2009.

[21] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016.

[22] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012.

[23] N. Gerhard, T. Ollmann, S. R. Bulo, and P. Kontschieder. The Mapillary Vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017.

[24] H. Grimmett, M. Buerki, L. Paz, P. Pinies, P. Furgale, I. Posner, and P. Newman. Integrating metric and semantic maps for vision-only automated parking. In *ICRA*, 2015.

[25] J. Guo, U. Kurup, and M. Shah. Is it safe to drive? an overview of factors, challenges, and datasets for driveability assessment in autonomous driving. *arXiv:1811.11277*, 2018.

[26] Q. Ha, K. Watanabe, T. Karasawa, Y. Ushiku, and T. Harada. Mfnet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes. In *IROS*, 2017.

[27] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[28] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[29] N. Homayounfar, W.-C. Ma, S. Kowshika Lakshmikanth, and R. Urtasun. Hierarchical recurrent attention networks for structured online maps. In *CVPR*, 2018.

[30] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang. The apolloscape open dataset for autonomous driving and its application. *arXiv:1803.06184*, 2018.

[31] H. Jung, Y. Oto, O. M. Mozos, Y. Iwashita, and R. Kurazume. Multi-modal panoramic 3d outdoor datasets for place categorization. In *IROS*, 2016.

[32] J. Kim, J. Choi, Y. Kim, J. Koh, C. C. Chung, and J. W. Choi. Robust camera lidar sensor fusion via deep gated information fusion network. In *IVS*, 2018.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[34] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander. Joint 3d proposal generation and object detection from view aggregation. In *IROS*, 2018.

[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 *CVPR*, 2019.

[36] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *ECCV*, 2018.

[37] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents `http://apolloscape.auto/tracking.html`. In *AAAI*, 2019.

[38] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000 km: The oxford robotcar dataset. *IJRR*, 2017.

[39] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka. 3d bounding box estimation using deep learning and geometry. In *CVPR*, 2017.

[40] L. Neumann, M. Karg, S. Zhang, C. Scharfenberger, E. Piegert, S. Mistr, O. Prokofyeva, R. Thiel, A. Vedaldi, A. Zisserman, and B. Schiele. Nightowls: A pedestrians at night dataset. In *ACCV*, 2018.

[41] A. Patil, S. Malla, H. Gang, and Y.-T. Chen. The H3D dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. In *ICRA*, 2019.

[42] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from RGB-D data. In *CVPR*, 2018.

[43] A. Rangesh and M. M. Trivedi. Ground plane polling for 6dof pose estimation of objects on the road. In *arXiv:1811.06666*, 2018.

[44] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[45] S. R. Richter, Z. Hayder, and V. Koltun. Playing for benchmarks. In *ICCV*, 2017.

[46] T. Roddick, A. Kendall, and R. Cipolla. Orthographic feature transform for monocular 3d object detection. In *arXiv:1811.08188*, 2018.

[47] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.

[48] K. Takumi, K. Watanabe, Q. Ha, A. Tejero-De-Pablos, Y. Ushiku, and T. Harada. Multispectral object detection for autonomous vehicles. In *Proceedings of the on Thematic Workshops of ACM Multimedia*, 2017.

[49] S. Wang, M. Bai, G. Mattyus, H. Chu, W. Luo, B. Yang, J. Liang, J. Cheverie, S. Fidler, and R. Urtasun. Torontocity: Seeing the world with a million eyes. In *ICCV*, 2017.

[50] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, 2019.

[51] Z. Wang, W. Zhan, and M. Tomizuka. Fusing birds eye view lidar point cloud and front view camera image for 3d object detection. In *IVS*, 2018.

[52] L. Woensel and G. Archer. Ten technologies which could change our lives. *European Parlimentary Research Service*, 2015.

[53] C. Wojek, S. Walk, and B. Schiele. Multi-cue onboard pedestrian detection. In *CVPR*, 2009.

[54] B. Xu and Z. Chen. Multi-level fusion based 3d object detection from monocular images. In *CVPR*, 2018.

[55] D. Xu, D. Anguelov, and A. Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *CVPR*, 2018.

[56] B. Yang, M. Liang, and R. Urtasun. HDNET: Exploiting HD maps for 3d object detection. In *CoRL*, 2018.

[57] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. BDD100K: A diverse driving video database with scalable annotation tooling. *arXiv:1805.04687*, 2018.

[58] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *SPL*, 23(10), 2016.

[59] S. Zhang, R. Benenson, and B. Schiele. Citypersons: A diverse dataset for pedestrian detection. In *CVPR*, 2017.