Ultra-Sonic Sensor based Object Detection for Autonomous Vehicles

Tommaso Nesti[1], Santhosh Boddana[2], Burhaneddin Yaman[1]
[1]Bosch Center for Artificial Intelligence, USA [2]Bosch Center for Artificial Intelligence, India
{tommaso.nesti, burhaneddin.yaman }@us.bosch.com, santhosh.boddana@in.bosch.com

Abstract

Perception systems in autonomous vehicles (AV) have made significant advancements in recent years. Such systems leverage different sensing modalities such as cameras, LiDARs and Radars, and are powered by state-of-the-art deep learning algorithms. Ultrasonic sensors (USS) are a low-cost, durable and robust sensing technology that is particularly suitable for near-range detection in harsh weather conditions, but have received very limited attention in the perception literature. In this work, we present a novel USS-based object detection system that can enable accurate detection of objects in low-speed scenarios. The proposed pipeline involves four steps. First, the input USS data is transformed into a novel voxelized 3D point cloud leveraging the physics of USS. Next, multi-channels Bird Eye's View (BEV) images are generated via projection operators. Later, the resolution of BEV images is enhanced by means of a rolling-window, vehicle movement-aware temporal aggregation process. Finally, the image-like data representation is used to train a deep neural network to detect and localize objects in the 2D plane. We present extensive experiments showing that the proposed framework achieves satisfactory performance across both classic and custom object detection metrics, thus bridging the usecase and literature visibility gap between USS and more established sensors.

1. Introduction

Object detection (OD) is one of the most significant perception task for autonomous vehicles as the information from OD is directly used for variety of fundamental tasks such as changing lanes, detecting traffic signals and road signs, and informing planning decisions [12, 14, 15]. Thus, OD models that are robust and can adapt to change in the environment are needed for reliable deployment of autonomous vehicles [2, 13]. Rapid development of deep learning has enabled developing advanced image based object detection models capable of handling changes in the environment such as varying light conditions or object orientation [7, 10]. While image-based object detection models can generally work well, their performance degrades in cases where the scene is blocked by obstacles such as during foggy or snowy weathers. Thus, a variety of sensors such LiDAR, Radar and ultrasonic sensors (USS) can be used for object detection purposes [4, 5]. Each of these sensors have advantages and disadvantages under different circumstances. Particularly, LiDAR works well with long range and different illumination conditions, but suffers from from adverse weather conditions and high costs [8, 12]. While Radar works well with varying ranges, illumination and weather conditions, it suffers from low-resolution and near-range performance [12]. On the other hand, USS performs well with near-range, low-speed and varying illumination conditions, yet suffers from varying temperature and humidity due to sensing properties [21].

Recently, there has been a significant interest in developing LiDAR and Radar based object detection models as each of these sensor can tackle existing challenges in camera-based object detection systems [23]. The success of such models has led to development of fusion architectures that further enhance the accuracy of perception tasks in autonomous vehicles [2, 5]. On the other hand, the lack of research on USS has hindered development for USS based object detection. Given their resiliency and low-cost compared to other sensors [21], development of USS based models can significantly contribute to the accuracy and safety of perception tasks for autonomous vehicles.

To this end, we present an end-to-end USS-based object detection framework. The proposed approach transforms input data from the sensors into USS-based bird's eye view images for training state-of-the-art object detector models. The main contributions of this work are listed as follows:

- A framework for transforming input tabular USS data into voxelized 3D point cloud.
- A novel USS data representation approach projecting the voxelized 3D point cloud data into multi-channels bird eye's view (BEV) images.
- A point cloud temporal aggregation approach to enhance the resolution of BEV images.
- A object detection pipeline leveraging the new representation and state-of-art computer vision models to

localize objects in the BEV plane, showing the feasibility and versatility of USS for autonomous driving.

2. Related Work

In recent years, object detection for perception tasks in autonomous vehicles has significantly progressed thanks to immense amount of research using camera sensor datasets. While cameras have been the mainstream sensor for developing various object detection models [8, 11, 18], there has also been numerous work on LiDAR and Radar perception [5, 22]. The data acquired from LiDAR sensors are 3D point clouds. Thus, 3D object detection networks has been a natural choice for point cloud based object detection. While initial works focused on manually crafted feature representations [3], more recent works have removed dependency on such hand-crafted features. These feature-learning oriented end-to-end trainable networks can be roughly categorized into two main categories, namely grid-based and point based methods. Point based methods [17] directly process the features from raw point cloud data without any transformation. while grid based methods [24] transform the point clouds into 3D voxels or 2D bird's eye view. Point-based approaches have more computational cost but have higher accuracy due to direct processing of point clouds. Grid-based approaches are computationally more efficient, but suffer from information loss due to transformations.

Due to their relatively low cost and suitability for near range detection, there has been interest in using ultrasonic sensors for certain AV applications. In particular, averaging and majority voting based distance estimation algorithms have been proposed for curb detection and localization [19]. In another line of work, capsule neural networks have been used for height classification [16]. However, to the best of our knowledge, there is no research on ML-based object detection using ultrasonic sensor data. Despite this, approaches developed for other sensors can be extended to USS. In the following, we adapt the grid-based 2D bird's eye view transformation approach explored in [1] to USS data, and leverage one-stage object detection model to meet the stringent time requirement in autonomous vehicles.

3. Methodology

3.1. Bird Eye's View Generation

Generation of BEV images from input USS data is done through a series of steps detailed below.

3.1.1 Ultra-sonic sensor data

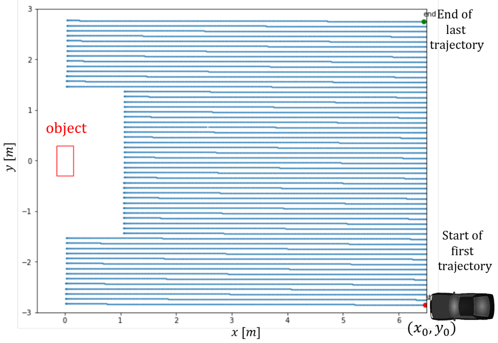


Figure 1. BEV visualization of the approach pattern of the bumper towards the object. Each approach trajectory $a = 0, \dots, D$ is a straight drive along the x-axis starting at $(x_0, y_0 + a \cdot 0.1)$ meters. Approaches are shifted along the y-axis by 0.1 meters to cover different reflection angles.

Ultrasonic sensors send out high-frequency sound waves to measure the distance to objects based on the measurement of the time of flight of the sonic wave from when it is emitted until the echo is received, comparing the object's echo amplitude against a threshold to detect an object [20]. The data used in this paper is collected using multiple ultrasonic sensors arranged over a car's front bumper. The bumper moves in a measurement room where different objects, such as poles, childdummies, bicycles and curbstones have been placed, following a predetermined approach pattern as in Figure 1. As the bumper moves, multiple sensors send out echoes simultaneously according to a fixed pattern, and to each sending sensor corresponds a unique receiving sensor. A set of such simultaneous echoes is called a *cycle*. For each echo in a cycle, the features of interest are the sender and receiver sensors, distance, and amplitude, as shown in Table 1.

Table 1. Example of features collected in 2 cycles of USS measurement.

time	cycle	echo	sender	receiver	dist.	amp.
t_1	c_1	e_{11}	s_{11}	r_{11}	d_{11}	a_{11}
		e_{12}	s_{12}	r_{12}	d_{12}	a_{12}
t_2	c_2	e_{21}	s_{21}	r_{21}	d_{21}	a_{21}

A limitation of this USS technology is that the data only include the distance travelled by the echo, but not the direction of measurement and angle of incidence, thus providing only the set of *potential* reflection points of an echo, rather than the exact (x, y, z) coordinates. Precisely, the *geometrical locus* of potential reflection points is provided, which in the case of an echo with same sender and receiver s and measured distance d can be approximated by the surface area of a 3D spherical cone with center in s, radius d, and cone angle dictated by the Field-of-View of the sensors. [1]

[1] In the case of a echo with different sender s and receiver r, the sphere is replaced by a 3D ellipsoid of revolution with sender and receiver sensors located at the focal points, with the major semi-axis being equal to d, the minor semi-axis uniquely determined by the sensors' coordinates, and the

This limitation is in contrast with the LiDAR data format, which is directly available as high-resolution 3D point clouds with exact (x, y, z) coordinates. This difficulty makes the development of USS-based object detection algorithms significantly harder compared to camera and LiDAR, and can help explain the very limited research on integrating USS in sensor fusion perception [6], with the majority of USS applications in autonomous driving being limited to parking assistance usecases. However, USS are particularly relevant for ultra-near range detection, and can complement other sensors by covering blind spots and providing necessary redundancy in case of adverse weather conditions [21]. In the following, we discuss how we circumvent this limitation to obtain point cloud-like input for USS, which in turn enable us to use established deep learning architecture to perform object localization, bridging the gap between USS and more expensive sensors.

3.1.2 3D Point cloud representation

Given an echo, the set of potential reflection points is a portion of the surface of a sphere as explained in Section 3.1. Next, we perform a discretization of the three-dimensional euclidean space into a 3D grid with voxel size 5-by-5-by-20 cm [2], and we discretize the sphere surface into this grid based on its intersections with the voxels. A 2D slice of the result is depicted in Figure 2. Each 3D grid cell can therefore be seen as a point in a discrete voxelized *3D point cloud*, where each point contains the coordinates of the cell's center, as well as information pertaining to the echoes intersecting the cell. Specifically, if a cell with center's coordinates $P = (x, y, z)$ is intersected by echoes $e_1, \ldots, e_k$, we are interested in the *number of echoes* num_ech_P intersecting the cell, and the list of corresponding amplitudes $(\text{amp}(e_i))_{i=1}^k$ and azimuth angles $(\text{azi}(e_i))_{i=1}^k$. [3]

In particular, the number of echoes intersecting a grid cell provides valuable information regarding the unknown coordinates of the reflection point: the higher this number (i.e., the red cell in Figure 2), the higher the likelihood that the reflection point lies in that grid cell due to a trilateration argument. This heuristic argument, while an approximation of reality due to the presence of spurious echoes (i.e. uneven ground) and noise in the recorded distances, proved to be useful in creating a meaningful point cloud representation of USS data. An example of a point cloud generated from a single cycle of echoes in shown in Figure 3.

The key difference between LiDAR and USS-based point cloud is that each point in a LiDAR point cloud already corresponds to the location of the laser beam reflection point, while each point in a USS point cloud corresponds to a *possible* location of the echo reflection point. We solve this problem by defining features associated to a point, such as the number of echoes in a cell, which allows to place more *weight* on the points with an higher likelihood of being close to the actual reflection point. For perception tasks, the weighted point cloud is then converted into a multi-channels BEV image encoding echoes as described in Section 3.1.3, which is ultimately used to train computer vision deep learning algorithms.

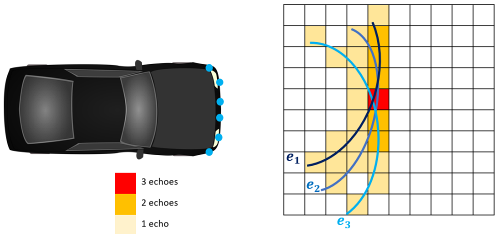


Figure 2. Example of voxel-based encoding of echoes e_1, e_2, e_3, shown as circular segments, in a 2D slice view. Sensors are depicted as blue dots on the car. The color intensity of a cell is proportional to the number of echoes intersecting it.

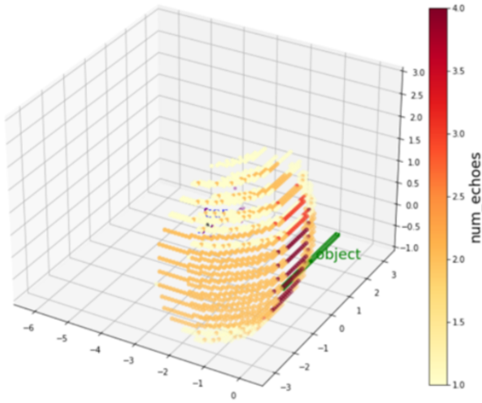


Figure 3. Example of voxelized 3D point cloud corresponding to a cycle with 4 echoes. Points refer to the centers of the 3D voxels, and are color-coded based the number of echoes in a cell. The higher-intensity points cluster around the location of the object (a curbstone), shown in green.

3.1.3 Bird eye's view image generation

The BEV image is obtained by projecting a single-cycle 3D point cloud into the xy plane, and consists of a 3-channel image encoding echoes intensity, amplitude, and azimuth angle information. An example is shown in Figure 4a. Specifically, given a cycle c and a (x^*, y^*) coordinate in the BEV image, corresponding to a 5-by-5 cm 2D grid cell, the three channels are defined as follows:

rotational symmetry being about the semi-axis of length d.

[2] This specific resolution was chosen based on a trade-off analysis between resolution and computational cost.

[3] Angles are between the viewing direction of the center of the sender and receiver and the segment from it to the cell center.

Echoes channel:

$$\text{ch\_ech}_{x^*,y^*}(c) = \max_z \text{num\_ech}_{x^*,y^*,z}(c)$$

The echoes channel contains the information on the number of echoes intersecting in a cell, which lies at the core of the heuristic geometric argument discussed in Section 3.1.2.

Range of amplitude/azimuth channel:

$$\text{max\_amp}_{x^*,y^*}(c) = \max\left(\bigcup_z \bigcup_{e \in c,\, e \text{ crosses } (x^*,y^*,z)} \text{amp}(e)\right)$$

$$\text{min\_amp}_{x^*,y^*}(c) = \min\left(\bigcup_z \bigcup_{e \in c,\, e \text{ crosses } (x^*,y^*,z)} \text{amp}(e)\right)$$

$$\text{ch\_amp}_{x^*,y^*}(c) = \text{max\_amp}_{x^*,y^*}(c) - \text{min\_amp}_{x^*,y^*}(c)$$

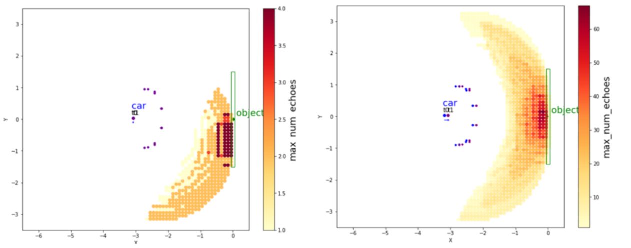
The range of azimuth channel is similarly defined, using the azimuth angles azi(e) rather than amp(e). The amplitude and azimuth channels encode non-trivial information about a grid cell's position relative to the sensors and the distribution of the amplitude values around it. Thus, they can improve the expressive power of the BEV representation by closing-in on specific parts of the set of potential reflection points compared to the echoes channel alone.

It can be helpful to interpret the three channels defined above as a USS-custom variant of the usual RGB channels in images. Even though deep learning architectures were designed with RGB images as input, they can be used with arbitrary spatially invariant 2D input to enable object detection. We note the framework described in this paper can easily be extended to handle additional channels.

3.1.4 Point cloud temporal aggregation

While single cycle point cloud data can be directly transformed into BEV for perception tasks, the resulting image is not sufficiently informative for localization purposes, due to the small number of echoes emitted during a cycle. In Figure 3 the number of echoes ranges from 1 to 4, thus not providing sufficient resolution for computer vision tasks.

In order to tackle this issue, we perform a *temporal aggregation* of 2D BEV data across a specified range of cycles. Aggregation is performed in a rolling window fashion using past data. Specifically, let $c_{t_1}, c_{t_2}, \ldots, c_{t_N}$ be the sequence of cycles in our measurement, corresponding to timestamps $t_1, t_2, \ldots, t_N$, and let K be an integer. At time t_i, we aggregate the K past cycles $c_{t_{i-K+1}}, \ldots, c_{t_i}$ into one single image by first offsetting each cycle's BEV by the amount needed to match the car's position at the final cycle t_i, and then superimposing the images as described in the channel definition 3.1.3. The exact offset calculation is possible because objects are *static* and the speed and direction of the car is known. The resulting channel-dependent aggregation strategy detailed as follows:



(a) Single-cycle BEV image.

(b) Temporally-aggregated BEV images using 1 second worth of cycles.

Figure 4. BEV projections of voxelized 3D point cloud. Points are color-coded based on the ch_ech(c, K) channel, with $K = 1$ (a) and $K = 32$ (b). Note the larger value range in (b), resulting in higher image resolution.

Echoes channel (aggregated): Aggregation is performed via sum operation over all cycles.

$$\text{ch\_echoes}_{x^*,y^*}(c_{t_i}, K) = \sum_{j=i-K+1}^{i} \text{max\_ech}_{x^*,y^*}(\tilde{c}_{t_j}).$$

where $\tilde{c}_{t_j}$ refers to the cycle c_{t_j} information after accounting for the offset. Note that $\tilde{c}_{t_i} = c_{t_i}$ since there is no offset at the final cycle c_{t_i}.

Range of amplitudes/azimuth channels (aggregated): Aggregation is performed by taking the difference between the cycle-wise max and min of the single-cycle amplitude and azimuth max and min, respectively. More formally, range of amplitudes can be formulated as

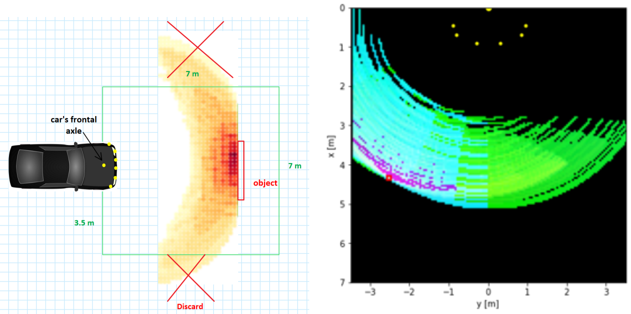
$$\begin{aligned}\text{ch\_amp}_{x^*,y^*}(c_{t_i}, K) = &\max_{j=i-K+1}^{i} \text{max\_amp}_{x^*,y^*}(\tilde{c}_{t_j}) \\ &- \min_{j=i-K+1}^{i} \text{min\_amp}_{x^*,y^*}(\tilde{c}_{t_j}).\end{aligned}$$

The same formulation applies to range of azimuth ch_azi channel. Finally, inference happens at cycle t_i, the end cycle of each aggregated sample. Figure 4b shows the point cloud obtained by aggregating $K = 32$ cycles, which is roughly equal to 1 second worth of measurements.

3.1.5 Normalization and data transformation

The BEV data is further passed through a normalization and preprocessing process prior to creating the train/test split. First, time-aggregated BEV images are cropped into a FoV of interest for USS applications, defined as a 7-by-7 meters square as shown in Figure 5a. Next, the channel values in the cropped BEV images are normalized in the range

$[0, 255]$, to account for the large variance in pixels' intensity across different time-aggregated samples. Finally, the coordinates of the ground truth bounding boxes are discretized to match the resolution of the 2D BEV images, with 1 pixel corresponding to a 5-by-5 grid cell. An example of a finalized training BEV image is provided in Figure 5b.



(a) BEV image cropping procedure. Sensors are shown as yellow dots on the car.

(b) Finalized BEV image for a pole object after normalization and additional preprocessing, $K = 32$.

3.2. Inference Framework

The resulting BEV images are used as input data for our deep learning object detection algorithm in 2D space.

3.2.1 Object Detection Framework

We adopt the single-shot detector (SSD) framework [11] to perform object detection using the USS BEV images generated as in Section 3.1.3. The architecture is composed of a backbone network which is responsible for computing a convolutional feature map over an entire input image, and a SSD head consisting of convolutional layers responsible for object classification and bounding box regression on the backbone's output. In this paper we focus on detection and localization and not on object classification, so we use only two classes for the classification layer, namely *object* and *no object*. Finally, the output of the model for each test image is the set of predicted bounding boxes together with the corresponding confidence score in $[0, 1]$, which quantifies the likelihood of the box containing an object.

3.2.2 Post-processing

In order to obtain accurate 2D bounding boxes predictions we perform several post-processing steps over the raw output of the network. First, we perform standard non-maximum suppression (NMS) to reduce the number of redundant and overlapping boxes, which is controlled by IoU and confidence score thresholds. While NMS is usually enough for most image-based detection tasks, the nature of USS data demands additional filtering steps to reduce the number of output boxes.

Specifically, the USS-based BEV encodes the set of potential reflection points of an echo (the echo's locus), rather than the actual reflection points. Therefore, the resulting BEV image will have a relatively high number of small, but non-zero pixels, corresponding to regions of the image far away from a reflection point which are however crossed by an echo's locus (for example, the low-intensity pixels in the left of Figure 6). At the same time, regions closer to the ground truth object tend to have higher-value pixels, but the distribution of values varies significantly across frames, due to different location, shape and material of the objects. As a consequence, the SSD model will generally predict many low-confidence boxes in regions without objects (which can safely be removed by NMS), but the confidence scores for true positives boxes can vary significantly. Consequently, blanket removal of boxes based on a low absolute confidence threshold in NMS will discard true positives as exemplified in Figure 6, where the low score true positive box for might be filtered out, while setting a high threshold might result in too many false positives boxes.

For this reason, we add a post-processing step based on a relative confidence threshold to discard remaining boxes post-NMS based on relative delta score between predicted boxes, on a per-frame basis. Finally, given a test BEV frame generated using K cycles $c_{t_{i-K+1}}, \ldots, c_{t_i}$, the output of the network is a minimal set of bounding boxes and confidence scores, representing the predicted location of the objects in the scene relative to the position of the car at time t_i.

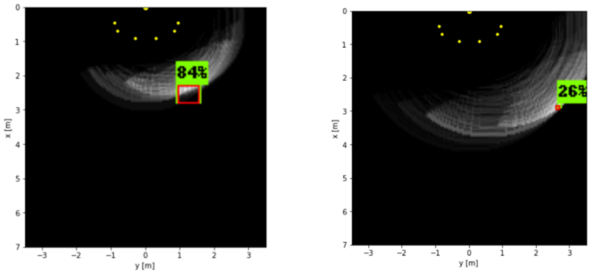


Figure 6. Sample of predictions for a childdummy (left) and a pole (right) after post-processing, with ch_ech channel shown in gray-scale. Ground truth and predicted boxes are shown in red and green, respectively.

4. Object Detection Experiments

In this section, we describe the dataset preparation, the experiment set up, the performance evaluation metrics and the results obtained for the USS-based object detector.

4.1. Dataset and Evaluation Setup

Train/Test split. Given the rolling-window nature of the temporal aggregation process described in Section 3.1.4, each BEV image belonging to the same approach trajectory (each blue line in Figure 1) overlaps with some of the preceding images. Moreover, these samples will be correlated to each other due to the similar reflection angles. To avoid data contamination between training and test set, we create our train/test split by assigning all samples belonging to the same trajectory to either the training set or the test set, in a $70/30$ proportion. Additional preprocessing includes removing samples spanning a different temporal duration than intended (i.e. outside $[1-\epsilon, 1+\epsilon]$ seconds for $K = 32$ cycles) as well as samples with zero-value entries in all channels, corresponding to anomalies in the data collection process. After such process, we end up with 86K and 39.5K images for training and testing, respectively.

Evaluation metric. For evaluation, we have used mAP$_{50}$, the mean average precision at intersection over union (IoU) value of 0.5, as well as the overall mAP averaged over 10 IoU thresholds between 0.5 and 0.95 based on the COCO implementation [9]. While these metrics have been the standard for camera based object detection models, they were unable to fully capture the prediction quality in several scenarios, where the object size in the grid-based BEV representation can be as low as 2 pixels, such as poles and curbstones. Such predictions can have low IoU value despite a satisfying quality of the localization accuracy, resulting in the mAP being an over-conservative metric.

Thus, in addition to these commonly used metrics, we have developed a customized key performance indicator (KPI) metric accounting for additional indicators of detection quality. First, we pair each predicted box to a ground truth box based on IoU levels, and we compute the IoU, area similarity and distance scores for each pair. The area similarity score is the minimum between the ratios of area between ground truth and predicted box, and vice versa. The distance score is measured by taking the euclidean distance between the center of the boxes, and then applying a transformation of the type $e^{-\alpha x}$ to scale the score in the range $[0, 1]$, where α is set based on a empirical study. Next, the custom KPI for a predicted box is obtained by taking the average of IoU, area and distance scores. Finally, the overall KPI is obtained by taking the average of all KPIs weighted by the corresponding bounding box's confidence score, after penalizing for false positives and missed detections.

4.2. Implementation Details

We have used the network architecture described in Section 3.2.1 using ResNet-50 as backbone. Specifically, weights are initialized using transfer learning from a pre-trained model on COCO dataset [9]. Despite the model being pre-trained for RGB images, the initialization proved helpful for USS-based images as well, thanks to the spatial-invariant nature of the representation. Our default model is trained with a batch size of 32, 50000 steps, stochastic gradient descent (SGD) with a cosine learning rate decay with base and cosine learning rate equal to $4 \cdot 10^{-2}$ and $1.33 \cdot 10^{-2}$, respectively, and 4000 warm-up steps. The input images are resized to 640×640 pixels while keeping the height/width ratio of the original images fixed, and a temporal aggregation of $K = 32$ cycles is used. We used weighted smooth ℓ_1 and weighted sigmoid focal loss for localization and classification, respectively. For data augmentation, we used random horizontal flip and random cropping.

4.3. Main Results

In this section, we present the key results obtained with the proposed USS framework.

General performance. Our default model's overall and object-level performance, evaluated over 7 different classes of objects, is reported in Table 2. The default model reaches a mAP$_{50}$ of 75.82 and a custom KPI value of 75.52. We further analyze the object-level performance based on custom KPI, with objects such as bicycle, childdummy, toy car and pole scoring the highest, whereas speedbump posed the most challenges for USS-based detection. This can possibly be attributed to the smooth round shape of speedbumps resulting in a large variance of echoes reflection angles. We note that curbstones, although having a similar height as speedbumps, are localized with better accuracy. An analysis of the impact of object shapes and materials on detection performance is a possible direction for future work.

Baseline. The absence of prior research on USS-based object detection imposes significant limitations on the ability to make direct comparisons. Nonetheless, to establish a baseline for comparison purposes, we have generated a baseline from our study. In particular, the baseline comparison methodology uses only the basic channel ch_ech, and does not have any temporal aggregation. For fair comparisons, we have used the same architecture described in 4.2 for both methodologies. Table 2 shows the USS based object detection performance of both baseline and default model. Multi-channel, temporally aggregated model shows significant improvement compared to the single-channel, non-aggregated baseline, in terms of both mAP and custom KPI. This result confirms the importance of increasing the resolution of the BEV representation by means of the temporal aggregation procedure described in Section 3.1.4.

Qualitative results. Figure 7 shows representative object detection results with and without temporal aggregation and additional channels. The baseline model fails to correctly detect and localize, whereas model trained on temporally aggregated multi-channels images show improved performance in terms of both detection and localization. Besides qualitative samples, we have studied how the distance from the object and the reflection angle impact performance in Figure 8, which shows the custom KPI at a more granular level. Each square represents the KPI value averaged over all test frames for which the center of the object's ground

Table 2. Overall and object-level performance of default model (3 channels, temporal aggregation with $K = 32$) vs baseline (1 channel, no aggregation).

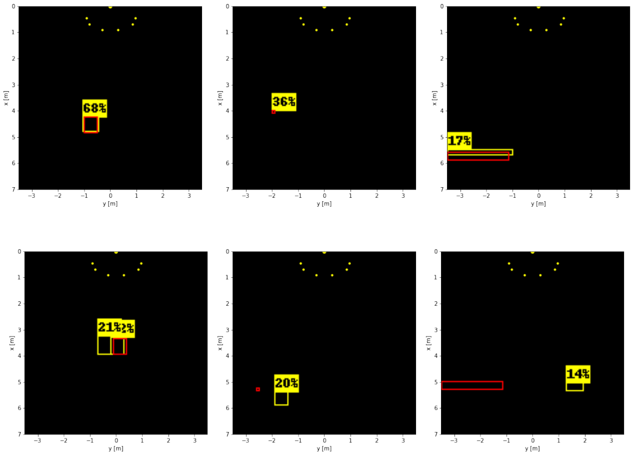
Model	mAP_{50}	mAP	New KPI							
			Overall	Bike	Childummy	Toy Car	Pole	Curbstone	Wall	Speedbump
Ours (default)	**75.82**	**40.65**	**75.53**	86.53	79.6	74.96	71.83	69.76	68.26	49.12
Baseline	18.88	7.03	47.17	53.62	46.9	43.86	37.05	47.36	46.4	31.67

Table 3. Object detection performance for different variants of the model.

Model	N. Steps	Resolution	Optimizer	Learning Rate	Data Augmentation	mAP_{50}	mAP
Ours (default)	50K	640 × 640	SGD	cosine decay	✓	75.82	40.65
Long	**100K**	640 × 640	SGD	cosine decay	✓	**77.62**	**42.87**
High Resolution	50K	**1024 × 1024**	SGD	cosine decay	✓	73.06	38.49
Momentum	50K	640 × 640	**SGD w/t momentum 0.89**	cosine decay	✓	71.93	34.71
Adam	50K	640 × 640	**Adam**	**1e-5**	✓	62.87	31.94
Fixed LR	$50K$	640 × 640	SGD	**1e-5**	✓	56.41	24.24
No augmentation	$50K$	640 × 640	SGD	cosine decay	×	59.30	23.6

Table 4. Object detection performance using different channels combinations to construct the BEV image inputs.

ch_ech	ch_amp	ch_azi	mAP_{50}	mAP	Custom KPI
✓	✓	✓	**75.82**	**40.65**	**75.53**
✓			71.69	37.74	75.44
	✓		49.89	20.12	68.16
		✓	50.23	20.22	68.59
✓	✓		71.76	36.37	74.19
✓		✓	73.03	38.75	74.59
	✓	✓	56.59	23.81	68.98



(a) Childdummy (b) Pole (c) Speedbump

Figure 7. Example of object detection results with default model (top) and baseline (bottom) for different objects and distances. Ground truth and predicted boxes are shown in red and yellow, respectively. The BEV channels are not shown to improve the visibility of the predictions.

truth bounding box lies within the square. [4] We note that our default model performs well overall, and is particularly accurate when objects are located in front of the car and $<$ 3 meters from the bumper, and tend to become slightly more inaccurate as the distance increase and the objects are further away from the frontal field of view (as can be appreciated in the speedbump example). This qualitative pattern agrees with the common wisdom that ultrasonic sensors are particularly useful for near-range detection and can help cover other sensors' blind spots [21]. Note how the single-channel, non time-aggregated baseline performs significantly worse, but still shows the same qualitative pattern of relative performance for different distances and angle.

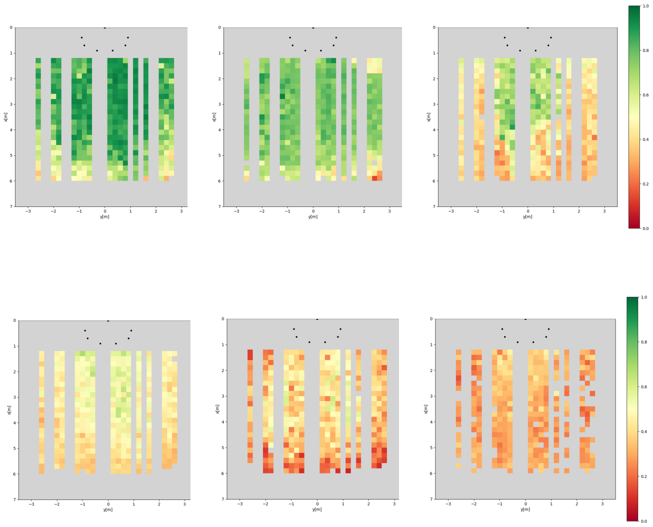
4.4. Ablation Studies

In this section, we validate the effect of each design choice and hyperparameter setting.

Channels Usage. A study on channels information impact on detection accuracy is carried out in order to understand the expressive power of each channel. For this analysis, we have trained our default model separately on single-channel images as well as on images with different channel combinations. As can be seen in Table 4, the most relevant information is contained in the max echoes channel, confirming the heuristic intuition behind the channel's definition as in Section 3.1.3. Conversely, the amplitude and azimuth angle channels have less expressive power and lead to sub-par performances, either taken alone or in combination. Finally, we note that adding amplitude and azimuth information on top of the main echoes channel improves the performance, if only incrementally, with the model trained on all three channels achieving the best performance among all the combinations. This suggests that including additional channels, leveraging domain knowledge of the underlying USS physics, can further enhance the performance.

Model selection and Hyperparameter Tuning In Table 3 we present several modifications done to the default model described in 4.2. For each variant, we changed one single hyperparameter while keeping the others the same.

[4] The empty squares refers to training trajectories excluded from testing.



(a) Childummy (b) Pole. (c) Speedbump.

Figure 8. OD performance for each relative position between car and object, for default (top) and baseline (bottom) models. Each square is 20-by-20 cm and is color-coded based on the avg. KPI for that relative position.

a) *Optimizer selection*: We compared Adam optimizer with Stochastic Gradient Descent (SGD) optimizer. For SGD, we conduct experiments with and without momentum. The results presented in Table 3 shows that OD model with SGD achieves significantly improved performance compared to model trained with Adam optimizer. SGD without momentum provides further improvements compared to SGD with the momentum.

b) *Image size*: We compared the baseline 640×640 resolution with 1024×1024 using SGD optimizer without momentum. As opposed to other sensing modalities such as camera, the model performs slighlty better with decreased resolution. A possible explanation for this result is the fact that USS-based BEV images are low-resolution by construction, since the pixel size corresponds to a relatively large 5-by-5 cm square. This pixel size is lower bounded by the actual margin of error of ultrasonic sensors, and excessive artificial upscaling can hinder the model's ability to learn key features such as object boundaries.

c) *Training duration*: We compared the model performance across varying training steps. As expected, model performance slightly improves with higher number of steps.

d) *Learning rate*: We compared a cosine decay learning rate schedule with a fixed learning rate, and have observed that the cosine schedule achieves significantly better results.

e) *Data augmentation*: We compared the results of our model trained with and without our default data augmentation scheme (random cropping and horizontal flipping), observing that the augmentation does improve performance significantly, as is often the case for standard RGB images.

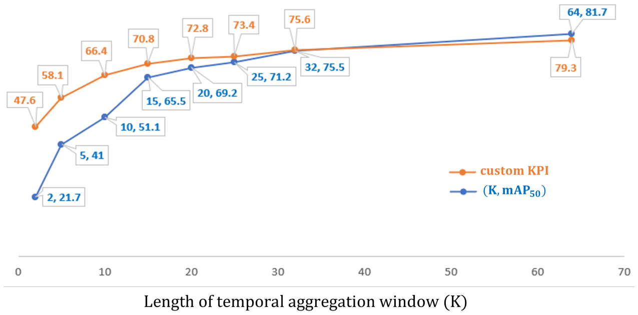


Figure 9. Impact of size of temporal aggregation window K on model performance. The diminishing return behavior can be used to inform the ideal size K for different usecases.

Effect of temporal aggregation Figure 9 shows the results of a sensitivity analysis with respect to different aggregation window sizes $K = 2, 5, 10, 15, 20, 25, 32, 64$, corresponding to temporal durations between 0.05 and 2 seconds. While both mAP$_{50}$ and custom KPI increase with K, the improvement tends to saturate and gives diminishing return after approximately $K = 16$, corresponding to ~ 0.5 seconds. This is especially important as longer temporal aggregation of 32 or 64 cycles (corresponding to 1 or 2 seconds) might not be feasible in real time scenarios with higher-speed and moving objects, where information from past ultrasonic cycles might become outdated quicker.

5. Conclusion

In this paper, we have proposed a novel ultrasonic sensor based object detection framework for autonomous vehicles. In particular, we developed an end-to-end framework where we first transformed input tabular USS data into voxelized 3D point cloud. Following, multi-channels bird's eye view images were obtained from the generated voxelized point clouds and used as input to a deep learning model based on the single-shot detector object detection architecture. We further proposed a channel-dependent temporal aggregation procedure for generating high quality images. Our extensive experiments have shown the feasibility of the proposed framework for object detection in autonomous vehicles using ultrasonic sensors, adding to the AV literature for sensors other than camera and LiDAR.

In the future, we plan to extend the approach to moving objects scenarios, perform object's height classification, and investigate how USS-based perception can be integrated in sensor-fusion pipelines. We hope that the findings of this study on ultrasonic sensor based object detection will serve as a catalyst for further research in the field of perception tasks for autonomous vehicles, and that the insights and guidelines provided in this study can serve as valuable baseline for future work in the area.

References

[1] Jorge Beltrán, Carlos Guindel, Francisco Miguel Moreno, Daniel Cruzado, Fernando Garcia, and Arturo De La Escalera. Birdnet: a 3d object detection framework from lidar information. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3517–3523. IEEE, 2018. 2

[2] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11682–11692, 2020. 1

[3] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. *Advances in neural information processing systems*, 28, 2015. 2

[4] Jamil Fayyad, Mohammad A Jaradat, Dominique Gruyer, and Homayoun Najjaran. Deep learning sensor fusion for autonomous vehicle perception and localization: A review. *Sensors*, 20(15):4220, 2020. 1

[5] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020. 1, 2

[6] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Gläser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2021. 3

[7] Hrag-Harout Jebamikyous and Rasha Kashef. Autonomous vehicles perception (avp) using deep learning: Modeling, assessment, and challenges. *IEEE Access*, 10:10523–10535, 2022. 1

[8] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia Computer Science*, 199:1066–1073, 2022. 1, 2

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 6

[10] Chun-Hao Liu and Burhaneddin Yaman. Object detection for autonomous dozers. *arXiv preprint arXiv:2208.08570*, 2022. 1

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016. 2, 5

[12] Enrique Marti, Miguel Angel De Miguel, Fernando Garcia, and Joshue Perez. A review of sensor technologies for perception in automated driving. *IEEE Intelligent Transportation Systems Magazine*, 11(4):94–108, 2019. 1

[13] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019. 1

[14] Darsh Parekh, Nishi Poddar, Aakash Rajpurkar, Manisha Chahal, Neeraj Kumar, Gyanendra Prasad Joshi, and Woong Cho. A review on autonomous vehicles: Progress, methods and challenges. *Electronics*, 11(14):2162, 2022. 1

[15] Scott Drew Pendleton, Hans Andersen, Xinxin Du, Xiaotong Shen, Malika Meghjani, You Hong Eng, Daniela Rus, and Marcelo H Ang Jr. Perception, planning, control, and coordination for autonomous vehicles. *Machines*, 5(1):6, 2017. 1

[16] Maximilian Pöpperli, Raghavendra Gulagundi, Senthil Yogamani, and Stefan Milz. Capsule neural network based height classification using low-cost automotive ultrasonic sensors. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 661–666. IEEE, 2019. 2

[17] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[19] Joon Hyo Rhee and Jiwon Seo. Low-cost curb detection and localization system using multiple ultrasonic sensors. *Sensors*, 19(6):1389, 2019. 2

[20] Francisca Rosique, Pedro J. Navarro, Carlos Fernández, and Antonio Padilla. A systematic review of perception system and simulators for autonomous vehicles research. *Sensors*, 19(3), 2019. 2

[21] Jorge Vargas, Suleiman Alsweiss, Onur Toker, Rahul Razdan, and Joshua Santos. An overview of autonomous vehicles sensors and their vulnerability to weather conditions. *Sensors*, 21(16), 2021. 1, 3, 7

[22] Ritu Yadav, Axel Vierling, and Karsten Berns. Radar+ rgb fusion for robust object detection in autonomous vehicle. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1986–1990. IEEE, 2020. 2

[23] De Jong Yeong, Gustavo Velasco-Hernandez, John Barry, and Joseph Walsh. Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors*, 21(6):2140, 2021. 1

[24] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 2