robotics

MDPI

Article

MonoGhost: Lightweight Monocular GhostNet 3D Object Properties Estimation for Autonomous Driving

Ahmed El-Dawy *, Amr El-Zawawi and Mohamed El-Habrouk

Faculty of Engineering, Electrical Power Engineering Department, Alexandria University, Lotfy El-Sied St. off Gamal Abd El-Naser, Alexandria 11432, Egypt; amr.elzawawi@yahoo.com (A.E.-Z.); eepgmme1@yahoo.co.uk (M.E.-H.)

* Correspondence: ahmed.dawy@alexu.edu.eg

Abstract: Effective environmental perception is critical for autonomous driving; thus, the perception system requires collecting 3D information of the surrounding objects, such as their dimensions, locations, and orientation in space. Recently, deep learning has been widely used in perception systems that convert image features from a camera into semantic information. This paper presents the MonoGhost network, a lightweight Monocular GhostNet deep learning technique for full 3D object properties estimation from a single frame monocular image. Unlike other techniques, the proposed MonoGhost network first estimates relatively reliable 3D object properties depending on efficient feature extractor. The proposed MonoGhost network estimates the orientation of the 3D object as well as the 3D dimensions of that object, resulting in reasonably small errors in the dimensions estimations versus other networks. These estimations, combined with the translation projection constraints imposed by the 2D detection coordinates, allow for the prediction of a robust and dependable Bird's Eye View bounding box. The experimental outcomes prove that the proposed MonoGhost network performs better than other state-of-the-art networks in the Bird's Eye View of the KITTI dataset benchmark by scoring 16.73% on the moderate class and 15.01% on the hard class while preserving real-time requirements.

Keywords: monocular 3D object detection; autonomous driving; robotics; perception; measurements

Citation: El-Dawy, A.; El-Zawawi, A.; El-Habrouk, M. MonoGhost: Lightweight Monocular GhostNet 3D Object Properties Estimation for Autonomous Driving. *Robotics* **2023**, *12*, 155. https://doi.org/10.3390/robotics12060155

Academic Editor: Yugang Liu

Received: 8 October 2023
Revised: 4 November 2023
Accepted: 6 November 2023
Published: 17 November 2023

1. Introduction

The development of driving assistance systems holds the possibility of reducing accidents, reducing environmental emissions, and easing the stress associated with driving [1–4]. Several levels of automation have been proposed, based on their technology capacities and human interaction [5,6]. The most widely known levels can be broken down into six groups [7]. Beginning with Level 0 (Driver-Only Level), the complete control of the vehicle, including steering, braking, accelerating, and decelerating, is completely under the control of the driver [8]. As the level increases from Level 1 to Level 3, the user interaction is reduced and the level of automation is increased [8]. In contrast to earlier levels of autonomous driving, Level 4 (High Driving Automation) and Level 5 (Full Driving Automation) attain fully autonomous driving, where the vehicle can be operated without the need for any driving experience or even a driving licence [9]. The difference between Level 4 and Level 5 is that autonomous vehicles categorized in Level 5 can drive entirely automatically in all driving domains and require no human input or interaction. Thus, Level 5 prototypes remove the steering wheel and the pedals; hence, the role of the driver is diminished to that of a mere passenger [10]. Consequently, there is a constraint on any vehicle equipped with a driving assistance system in order to evolve into a practical reality. This vehicle must be equipped with a perception system that enables high levels of awareness and intelligence necessary to deal with stressful real-world situations, make wise choices, and always act in a manner that is secure and accountable [5,11,12].

The perception system of autonomous cars performs a variety of tasks. The ultimate purpose of these tasks is to fully understand the vehicle's immediate surroundings with low latency so that the vehicle can make decisions or interact with the real world. Object detection, tracking, and driving area analysis are examples of such activities. 3D object properties detection is a significant area of study in the autonomous vehicles perception system [13]. Current 3D object detection techniques can be primarily classified into Lidar-based or vision-based techniques [14]. Lidar-based 3D object detection techniques are accurate and efficient but expensive, which limits their use in industry [15]. On the other hand, vision-based techniques [16] can also be categorized into two distinct groups: monocular and binocular vision. Vision-based perception systems are widely used due to their low cost and rich features (critical semantic elements within the image that carry valuable information and are used to understand and process the visual data). The most significant downside of monocular vision is that it cannot determine depth directly from image information, which may cause errors in 3D pose estimation in monocular object detection. The cost of binocular vision is higher than that of monocular vision, although it can provide more accurate depth information than monocular vision. Moreover, binocular vision yields a narrower visual field range, which cannot meet the requirements of certain operating conditions [17].

For the case of monocular vision systems, the camera projects a 3D point (defined in the 3D world's coordinate frame of the object) into the 2D image coordinate frame. This is a forward mathematical operation, which removes the projected object's depth information. Thus, the inverse projection of the point from the 2D image coordinate frame back into the 3D world's coordinate frame of the object is not a trivial mathematical task [18].

A 2D object detector block is fed with the 2D captured image and outputs the 2D coordinates of the object of interest defined in the 2D image coordinate frame as well as a cropped image of that particular object (inside the 2D bounding box). The target of the 3D perception system is to recover the object's 3D bounding box described in the 3D world's coordinate frame as well as its Bird's Eye View. The Bird's Eye View posture is predicted by neglecting the height of the object. A plethora of algorithms have been developed in the literature to perform this task. These algorithms will be discussed in the following section.

In this study, the proposed MonoGhost network, which is a lightweight deep learning-based technique, is proposed to estimate the object's 3D posture (Position and Orientation) and 3D dimensions from the monocular observed 2D bounding box (a rectangular or square-shaped region that encapsulates and outlines a specific object or region of interest within an image). The estimated object's orientation and 3D dimensions are then used to generate the Bird's Eye View of the detected object. The proposed MonoGhost network can use the state-of-the-art 2D object detectors [19–21], which can be enhanced to 3D object detectors by training an efficient lightweight feature extractor. Its extracted features are then used to predict the object's 3D bounding box orientation and dimensions. Taking into account the estimated 3D bounding box orientation and dimensions as well as the coordinates of the object's 2D bounding box, the object's Bird's Eye View bounding box can be generated. This paper is restricted to addressing only objects of class "Car".

The related work regarding the main methodologies of the 3D object detection are discussed in Section 2. The proposed MonoGhost network, which is explained in Section 3, provides the following contributions:

- A deep learning approach for estimating the Bird's Eye View bounding box of the detected object, depending on encoding the object's geometric and visual features using 1D convolution then fusing the encoded features to decode the object Bird's Eye View translation center.
- Preserving the stability of object's depth prediction for KITTI [22] hard object case without sacrificing the orientation prediction accuracy.
- Simple design of MonoGhost network, which is composed of low computational burden operations.
- Selection of an efficient lightweight, embedded-device-friendly feature extractor.

The experimental work of the proposed MonoGhost network, which explains the implementation details, the utilized dataset for training and benchmarking, and the training procedures, is discussed in Section 4. The success of the proposed network architecture is verified by experimental results on KITTI Bird's Eye View benchmark (https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=bev, accessed on 5 November 2023). It is compared to the most effective monocular 3D object identification methods currently available like PGD-FCOS3D [23], KM3D-Net [24], and SMOKE [25], as addressed in Section 5. Conclusions are presented in the final section of the paper.

2. Related Work

The various methods employed for 3D object detection can be classified into two distinct categories: conventional techniques or deep learning techniques. This section provides a comprehensive summary of the present 3D object detection methodologies as summarized in Figure 1. The 3D-data-based techniques in both categories achieve superior detection results over 2D-data-based techniques. However, they come with additional cost related to the sensor setup, as mentioned earlier. On the other hand, 2D-data-based techniques are cost efficient, which comes with sacrificing the depth estimation accuracy [15,17].

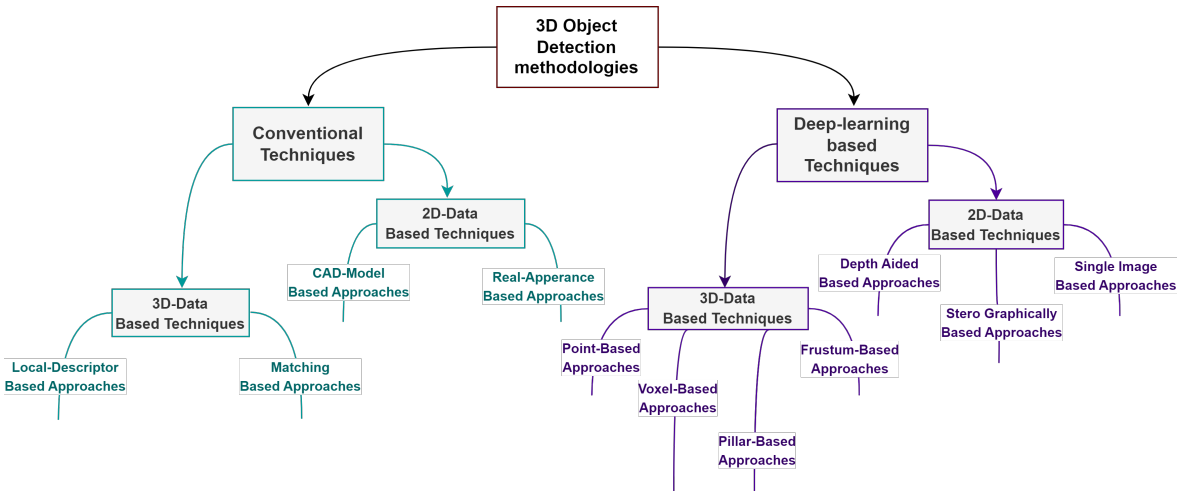

Figure 1. 3D object detection methodologies.

2.1. Conventional Techniques

Conventional techniques depend heavily on a prior knowledge of the object appearance to retrieve the essential suite of object features using hand-crafted feature descriptors, followed by the formation of a database by appropriately attributing features with the 3D models [26–28]. In order to provide quick detection results, the model database is organized and searched using effective indexing algorithms such as hierarchical RANSAC search, and voting is performed by using ranked criteria [29]. Other approaches depend on different hashing techniques like geometric hashing [26] and elastic hash table [30]. The conventional techniques can be categorized according to the input information into 3D-data-based techniques and 2D-data-based techniques [31].

2.1.1. 3D-Data-Based Techniques

With the advancement of sensor technology, more devices capable of capturing 3D environmental data, such as depth cameras and 3D scanners [32,33], are developed. When compared to 2D-data techniques, 3D-data techniques retain the object's genuine physical characteristics, which make them a superior choice to quantify the 6D pose [34]. The two primary categories of conventional 3D-data-based approaches are as follows.

Local-Descriptor-Based Approaches: These depend on the local descriptor and utilize an offline global descriptor applied to the model. This global descriptor needs to be ro-

tation and translation invariant. Then, the local descriptor is subsequently generated online and matched with the global descriptor. Super Key 4-Points Congruent Sets (SK-4PCS) [35] can be paired with invariant local properties of 3D shapes to optimize the quantity of data handled. The iterative Closest Point (ICP) [36] approach is a traditional method that can compute the pose transformation between two sets of point clouds. Another approach [37] establishes a correspondence between the online local descriptor and the saved model database by employing the enhanced Oriented Fast and Rotated BRIEF (ORB) [38] feature and the RBRIEF descriptor [39]. A different approach centered around semi-global descriptors [40] can be used to assess the 6D pose of large-scale occluded objects.

Matching-Based Approaches: The objective of matching-based approaches is to find the template in the database that is almost identical to the input sample as well as retrieving its 6D pose. An innovative Multi-Task Template Matching (MTTM) framework was proposed in [41] to increase matching reliability. It locates the closest template of a target object while estimating the masks of segmentation and the object's pose transformation. In order to enhance the data storage footprint and the lookup searching time, fused Balanced Pose Tree (BPT) and PCOF-MOD (multimodal PCOF) under optimum storage space restructuring was proposed in [42] in order to yield memory-efficient 6D pose estimation. A study on the control of micro-electro-mechanical system (MEMS) microassembly was conducted in [43] to allow for an accurate control to enable minimizing 3D residuals.

2.1.2. 2D-Data-Based Techniques

To measure the 6D pose of objects, a variety of features that are represented in the 2D image can be used. These features may include texture features, geometric features, color features, and more [31,44,45]. Speeded Up Robust Features (SURF) [46] and Scale-Invariant Feature Transform (SIFT) features [47] are considered the primary features that can be utilized for object pose estimation. Color features [48], as well as geometric features [49], can be utilized to enhance the pose estimation performance. The conventional 2D-data-based techniques can be further broken down according to the used matching template into CAD-model-based approaches and real-appearance-based approaches.

CAD-Model-Based Approaches: CAD-modelbbased approaches depend mainly on the rendered templates of CAD models. These approaches are suitable for industrial applications, because illumination and blurring have no effect on the rendering process [31]. A Perspective Cumulated Orientation Feature (PCOF) based on orientation histograms was proposed in [50] to estimate a robust object's pose. The Fine Pose parts-based Model (FPM) [51] was implemented to localize objects in a 2D image using the given CAD models. Moreover, the edge correspondences can be used to estimate the pose [52]. The multi-view geometry can be adapted in order to extract the object's geometric features [53], while the epipolar geometry can be used to generate the transformation matrix [54]. Some other approaches suggest visuals-based tracking like [55], which performs tracking approach based on CAD model for micro-assembly and like [56] which employs 6D posture estimation for end-effector tracking in a scanning electron microscope to enable higher-quality automated processes and accurate measurements.

Real-Appearance-Based Approaches: Despite the accuracy that can be achieved by employing 3D CAD models, reliable 3D CAD models are not always obtained. Under these conditions, real object images are used as the template. The 6D pose of an object can be measured depending on multi-cooperative logos [57]. The Histogram Of Gradients (HOG) [58] is an effective technique for improving the pose estimation performance. Another approach [59] suggested template matching in order to provide stability against small image variations. As obtaining sufficient real image templates is laborious and time-consuming, and producing images via CAD model is becoming simpler, approaches that depend on CAD models are much more popular rather than approaches that depend on real appearance [31].

The main downside of the conventional techniques is that they are very sensitive to noise and are not efficient in terms of computational burden and storage resources [60].

The performance of the 2D-data-based techniques heavily depends on the total number of templates, which impacts the pose estimation accuracy, as these approaches depend on the object appearance experience. The greater the number of templates, the more accurate the posture assessment [61,62]. Consequently, a large number of templates necessitates a significant amount of storage and search time [63]. Also, it is not practical to obtain 360° features of the model [64,65]. Because of how the 3D-data-based approaches function, the major limitation for these approaches is that they fail to function adequately whenever the object has a high level of reflected brightness [31]. Another problem is that the efficiency of these approaches is somewhat limited, as point clouds and depth images involve an enormous quantity of data, leading to a large computational burden [31].

2.2. Deep Learning-Based Techniques

Convolution Neural Networks (CNNs) are commonly used in 3D object detection methodologies based on deep learning strategies to retrieve a hierarchical suite of abstracted features from each object in order to record the object's essential information [66]. Unlike conventional 3D object detection techniques, which depend on hand-crafted feature descriptors, deep learning-based techniques utilize learnable weights that can be automatically tuned during the training phase [67]. Thus, these techniques provide more robustness against environmental variations. The deep learning techniques can also be categorized into 3D- and 2D-data-based techniques, as shown in Figure 1.

2.2.1. 3D-Data-Based Techniques

In recent years, LiDAR-based 3D detection [68–71] has advanced rapidly. LiDAR sensors collect accurate 3D measurement data gathered from the surroundings in the pattern of 3D points (x, y, z), at which x, y, z are each point's absolute 3D coordinates. LiDAR point cloud representations, by definition, necessitate an architecture that allows convolution procedures to be performed efficiently. Thus, deep learning-based techniques, which depend on LiDAR 3D detection, can be divided into Voxel-based approaches, Point-based approaches, Pillar-based approaches, and Frustum-based approaches [72,73].

Voxel-Based Approaches: These partition point clouds into similarly sized 3D voxels. Following that, for every single voxel, feature extraction can be used to acquire features from a group of points. The aforementioned approach minimizes the overall size of the point cloud, conserving storage space [73]. Voxel-based approaches, such as VoxelNet [74] and SECOND [75], augment the 2D image characterization into 3D space by splitting the 3D space into voxels. Other deep learning models like Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection [76], Afdet [77], Center-based 3D object detection and tracking [78], and Psanet [79] utilize the Voxel-based approach to perform 3D object detection.

Point-Based Approaches: These were introduced to deal with raw unstructured point clouds. Point-based approaches, such as PointNet [80] and PointNet++ [81] issue raw point clouds as input and retrieve point-wise characteristics for 3D object detection using formations such as multi-layer perceptrons. Other models like PointRCNN [71], PointRGCN [82], RoarNet [83], LaserNet [84], and PointPaiting [85] are examples of deep learning neural networks that adopt a Point-Based approach.

Pillar-Based Approaches: These structure the LiDAR 3D point cloud into vertical columns termed as pillars. Utilizing pillar-based approaches enables the tuning of the 3D point cloud arranging process in the x–y plane, removing the z coordinate, as demonstrated in PointPillars [86].

Frustum-Based Approaches: These partition the LiDAR 3D point clouds into frustums. The 3D object detector models that crop point cloud regions recognized by an RGB image object detector include SIFRNet [87], Frustum ConvNet [88], and Frustum PointNet [89].

2.2.2. 2D-Data-Based Techniques

RGB images are the main input for deep learning-based techniques that rely on 2D data. Despite the outstanding performance of 2D object detection networks, generating 3D bounding boxes merely from the 2D image plane is a substantially more complicated challenge owing to the lack of absolute depth information [90]. Thus, the 2D-data-based techniques for 3D object detection can be categorized according to the adopted method to generate the depth information into Stereographically Based Approaches, Depth-Aided Approaches, and Single-Image-Based Approaches.

Stereographically Based Approaches: Strategies in [91–95] process the pair of stereo images using a Siamese network and create a 3D cost volume to determine the matching cost for stereo matching using neural networks. The preliminary work 3DOP [96] creates 3D propositions by tinkering with a wide range of extracted features such as stereo reconstruction, and object size previous convictions. MVS-Machine [97] considers differentiable projection and reprojection for improved 3D volume construction handling from multi-view images.

Depth-Aided Approaches: Due to the missing depth information in single monocular image input, several researchers tried to make use of the progress in depth estimation neural networks. Previous research works [98–100] convert images into pseudo-LiDAR perceptions by harnessing off-the-shelf depth map predictors and calibration parameters. Then, they deploy established LiDAR-based 3D detection methods to output 3D bounding boxes, leading to lesser effectiveness. D4LCN [101] and DDMP-3D [102] emphasize a fusion-based strategy between image and estimated depth using cleverly engineered deep CNNs. However, most of the aforementioned methods that utilize off-the-shelf depth estimators directly pay substantial computational costs and achieve only limited improvement due to inaccuracy in the estimated depth map [103].

Single-Image-Based Approaches: Recently, several research works [13,104–107] employed only a monocular RGB image as input to 3D object detection. PGD-FCOS3D [23] establishes geometric correlation graphs between detected objects then utilizes the constructed graphs to improve the accuracy of the depth estimation. Some research works like RTM3D [108], SMOKE [25], and KM3D-Net [24] anticipate key points of the 3D bounding box as an adjacent procedure for establishing spatial information of the observed object. Decoupled-3D [109] presents an innovative framework for the decomposition of the detection problem into two tasks: a structured polygon prediction task and a depth recovery task. QD-3DT [110] presents a system for tracking moving objects over time and estimating their full 3D bounding box from an ongoing series of 2D monocular images. The association stage of the tracked objects depends on quasi-dense similarity learning to identify objects of interest in different positions and views based on visual features. MonoCInIS [111] presents a method that utilizes instance segmentation in order to estimate the object's pose. The proposed method is camera-independent in order to account for variable camera perspectives. MonoPair [112] investigates spatial pair-wise interactions among objects to enhance detection capability. Many recent research works depend on a prior 2D object detection stage. Deep3Dbox [113] suggests an innovative way to predict orientation and dimensions. M3D-RPN [114] considers a depth-aware convolution to anticipate 3D objects and produces 3D object properties with 2D detection requirements. GS3D [115] extends Deep3Dbox with a feature extraction module for visible surfaces. Due to the total loss of depth information and the necessity for a vast search space, it is not trivial to estimate the object's spatial position immediately [17]. As a result, PoseCNN [116] recognizes an object's position in a 2D image while simultaneously predicting its depth to determine its 3D position. Because rotation space is nonlinear, it is challenging to estimate 3D rotation directly with PoseCNN [17].

Although deep learning methods that rely on a single RGB image provide some appropriate projection-based constraints, they fail to achieve promising results due to an absence of depth spatial data [98,117].

3. MonoGhost Network

The proposed MonoGhost network can be categorized as a deep learning single image approach based on prior 2D bounding box detection. The mentioned approaches in the previous section that belong to the same category rely on a prior 2D detection stage. They fail to achieve promising depth results depending only on the projection constraint. Moreover, it is not easy to output a reliable depth estimation while maintaining good orientation performance [17].

The proposed MonoGhost relies on a features-fusion architecture to satisfy the projection constraint and output a stable object depth included in the object's Bird's Eye View. The proposed MonoGhost network, as will be presented in the remaining part of the section, can output steady object depth without sacrificing good orientation estimation performance while preserving real-time performance.

By relying on the assumption that a 3D bounding box's perspective projection should closely fit within its 2D observed bounding box, it is possible to build upon the accomplishments of previous efforts on 2D object detection for 3D bounding box estimation [118]. This is why MonoGhost was designed with the ability to be seamlessly integrated with any state-of-the-art 2D object detector. MonoGhost relies on Faster R-CNN [119] to provide the 2D bounding box coordinates for all the results obtained in this research.

As Figure 2 shows the angles, dimensions, and translations on an object model, the 3D bounding box could be described by its dimensions $\mathrm{D} = [d_x, d_y, d_z]$, center coordinates $\mathrm{T} = [t_x, t_y, t_z]^T$, and orientation R (θ, ϕ, α), which are characterized by azimuth (θ), elevation (ϕ), and roll (α) angles. For the assumed case of a flat ground, ϕ and α angles are safely considered zero. Moreover, by considering all the objects to be on the ground, it is also valid to fix the height of the object (t_y) to zero. Thus, it is sufficient to predict the Bird's Eye View posture.

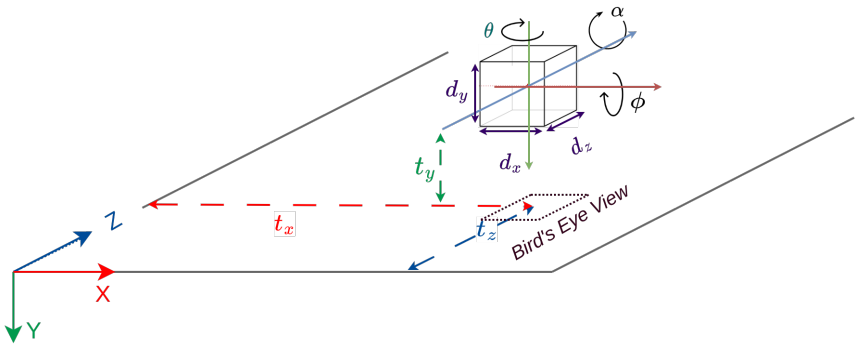

Figure 2. Diagram of the angles, dimensions, and translation on a 3D object model.

It is then required to calculate the projection of a 3D point $X_o = [X, Y, Z, 1]^T$, at which X, Y, Z are the point's 3D coordinates described in the world's 3D coordinate frame, which originates at the object's center T, into the 2D image coordinate frame, $x = [u, v, 1]^T$. This can be performed, as in [18,113,120], and as detailed in Equation (1), given the object's posture in the camera coordinate frame $(R, T) \in SE(3)$ and the intrinsic parameters of the camera, which are represented by matrix K:

$$x = K[\ R\ \ T\ ]X_o \tag{1}$$

Therefore, in order to recover the 3D coordinates of the object from 2D image, full representation of the object's geometric features (Dimensions and Orientation) must be estimated first. That is why the proposed MonoGhost network, shown in Figure 3, first estimates the orientation and dimensions of the object (through the Orientation-Dimensions Estimator block). These estimates are then joined with the object's 2D bounding box coordinates and the Guiding Depth, which represents the initial depth estimation of the object, to generate the required 3D Bird's Eye View posture (through the Bird's Eye View Center Estimator block). The proposed model learns to transform 2D detection from the

front camera view into a Bird's Eye View occupancy map. Consequently, the proposed MonoGhost network is composed of two main stages as presented in Figure 3:

- **First stage** (Orientation-Dimensions Estimator) accepts the cropped object image as well as the coordinates of the 2D detected bounding box, then extracts object visual features and outputs the object geometric features (orientation and dimensions).
- **Second stage** (Bird's Eye View Center Estimator) fuses the object's visual and geometric features with the 2D bounding box coordinates and the Guiding Depth to estimate the object Bird's Eye View bounding box center.

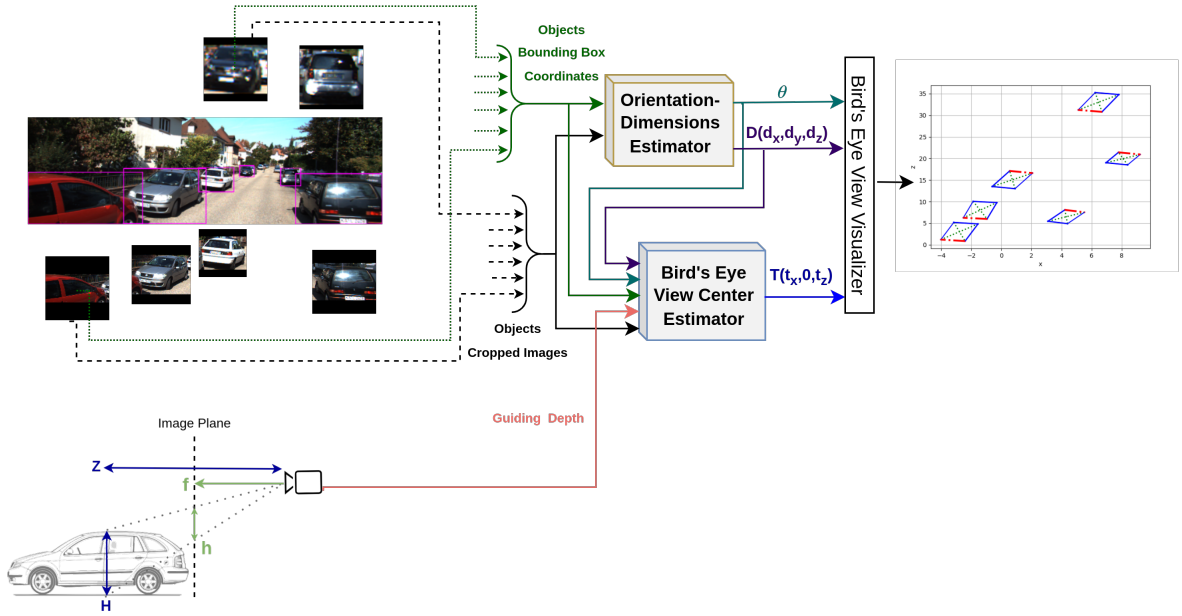

Figure 3. The general architecture of the proposed MonoGhost network is composed of two stages: the Orientation-Dimensions Estimator stage and Bird's Eye View Center Estimator stage.

3.1. Orientation Estimation

To estimate the global orientation of the object $R \in SO(3)$, it is insufficient to depend only on the 2D detection's contents, since the position of the 2D bounding box in the plane of the image is also unavoidable. Consider the rotation R(θ) with only azimuth as a parameter, θ (yaw).

Figure 4 illustrates an instance of a vehicle crossing the road. The vehicle's global direction θ does not alter, but its local direction θ_l which is estimated depending on the appearance of the vehicle in the 2D bounding box is changed. Thus, the global direction θ is calculated by adding the change in the local direction θ_l with respect to θ_{ray}, which represents the ray traversing the center of the cropped bounding box originating from the camera center. Since the direction of a ray at a given pixel can be easily calculated given intrinsic camera parameters, it is crucial to estimate the local orientation θ_l using the 2D bounding box's extracted features. Then, by adding the estimated local orientation θ_l and the ray direction to the center of the observed 2D bounding box θ_{ray}, the object's global orientation θ is calculated.

As shown in Figure 5, The Orientation-Dimensions Estimator utilizes the GhostNet [121] as a backbone to extract semantic visual feature of the object. Then the extracted features are shared between two main estimation heads. The first estimation head is the Local Orientation Estimation Head that estimates the object's local orientation θ_l. The estimated θ_l is then added to the calculated θ_{ray} in order to output the object's global orientation θ. The second estimation head is the Dimensions Estimation Head that outputs the 3D object's dimensions D $= [d_x, d_y, d_z]$.

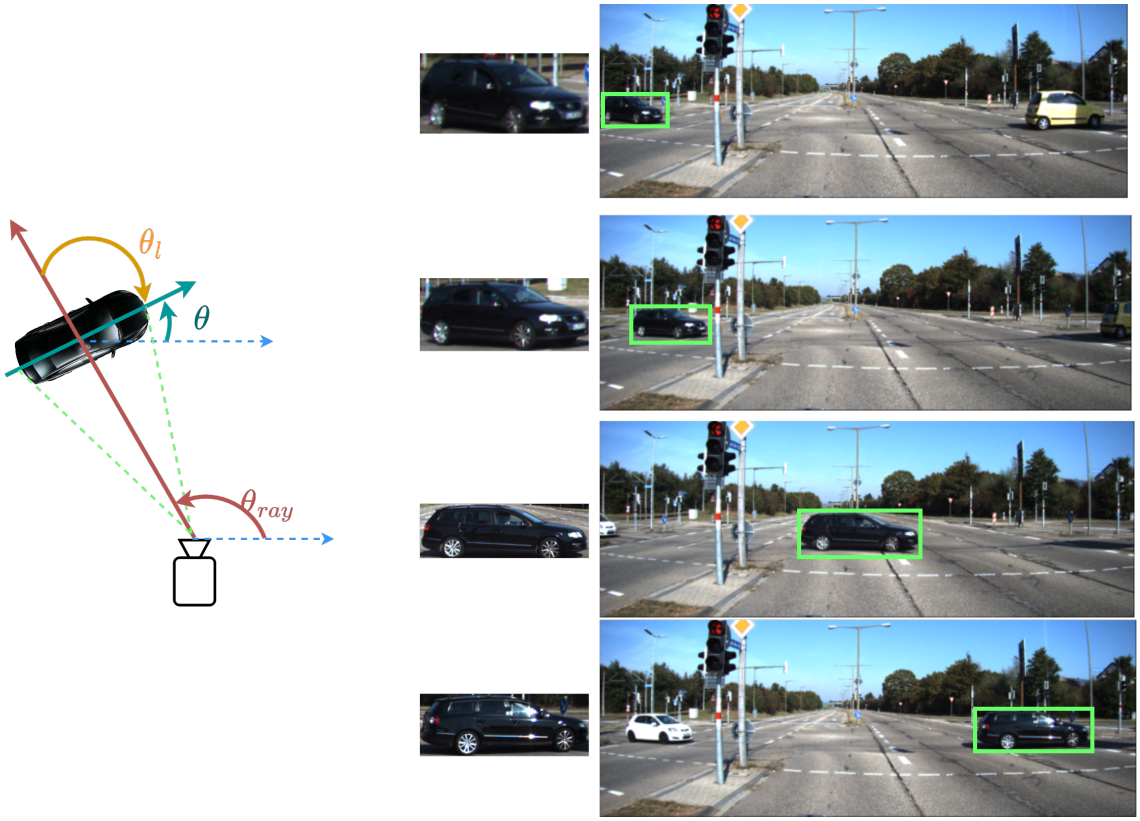

Figure 4. Orientation of the object θ is calculated as $\theta_{ray} + \theta_l$. The Orientation-Dimensions Estimator outputs θ_l, while θ_{ray} can be calculated with respect to the center of the object bounding box with the known camera's intrinsic parameters.

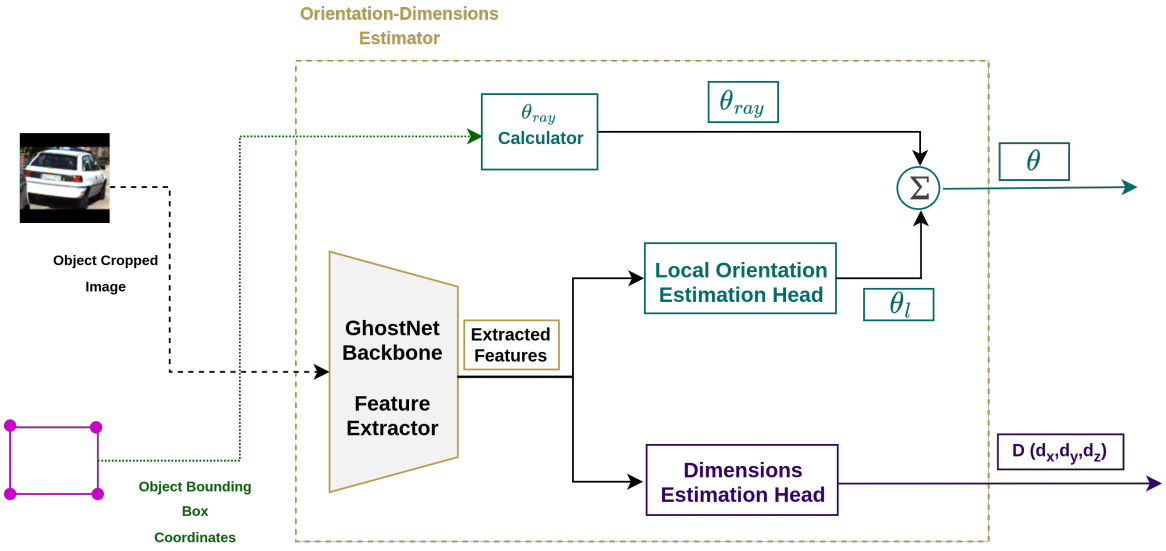

Figure 5. Orientation-Dimensions Estimator.

Object detectors such as YOLO [19] and SSD [122] partition the space of potential bounding boxes into a number of distinct modes known as anchor boxes, and then quantify the continuous offsets that must be adapted to each anchor box. Following a similar idea, the MultiBin architecture [113] can be used for local orientation estimation. It first discretizes the orientation angle space by dividing it into n overlapping bins. For each bin, the Local Orientation Estimation Head predicts both the score probabilities $score_i$ that the orientation angle is enclosed within the i^{th} bin and the residual correction angle that must

be added to the direction of this bin's center angle to yield the specified orientation angle. The residual rotation angle is denoted by two numbers: sine and cosine of that angle. As a result, each bin i has three outputs: $score_i$, $sine(\Delta\theta_i)$, and $cosine(\Delta\theta_i)$.

Consequently, the MultiBin orientation has a total loss of [113], as detailed in Equation (2):

$$L_\theta = L_{score} + \alpha \times L_{residual} \tag{2}$$

The softmax loss [123] of the scores for each bin determines the score loss L_{score}. The residual loss $L_{residual}$ tends to reduce the gap between the predicted angle and the ground truth angle. Therefore, the residual loss $L_{residual}$, is equivalent to maximizing cosine distance and can be calculated as explained in Equation (3) [113]:

$$L_{residual} = \frac{1}{n_{\theta^*}} \sum cos(\theta^* - c_i - \Delta\theta_i) \tag{3}$$

where n_{θ^*} is the number of bins that cover the ground truth angle θ^*. c_i is the angle of the i^{th} bin center. $\Delta\theta_i$ is the adjustment that must be made to the center of bin i.

As shown in Figure 6, in the MonoGhost network the orientation angle space is divided into two bins. Thus, the first branch in the Local Orientation Estimation Head has (1280, 256, 2) Fully Connected Network (FCN) units and outputs the bins scores, while the last branch is composed of (1280, 256, 2 × 2) FCN units and generates the residual correction angles sine and cosine.

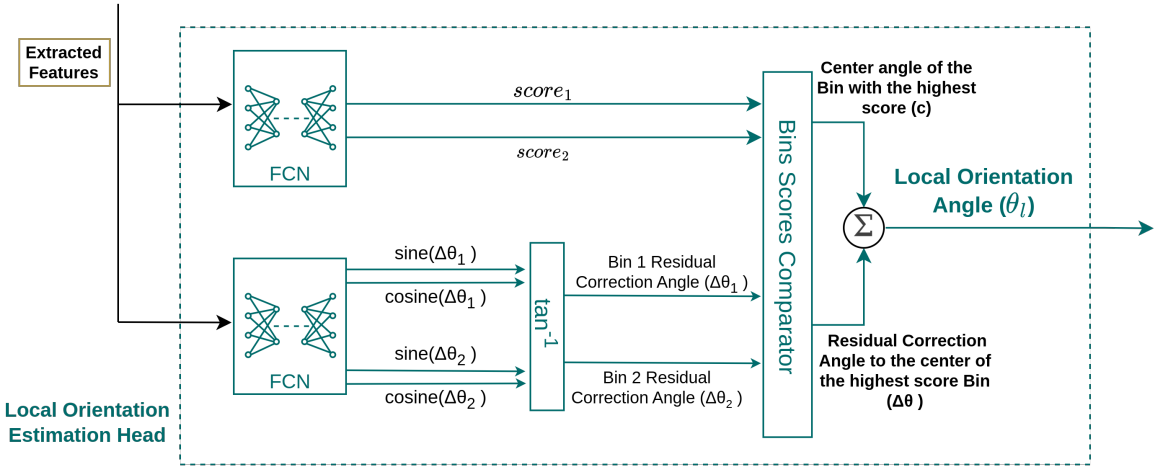

Figure 6. Local Orientation Estimation Head.

3.2. Dimensions Estimation

Cars, vans, trucks, pedestrians, cyclists, and buses are all separate categories in the KITTI dataset [22]. It is noted easily that the objects in each category have a similar shape and size. As an illustration, the dimensions variance for bicycles and cars have limits of centimeters. As a result, instead of depending on a discrete-continuous loss, such as the MultiBin loss, the L2 loss [124] can be used directly. For each dimension it is convenient to predict the deviation value from the average parameter value calculated over the training dataset. The dimensions estimation loss can be calculated as the explanation provided in Equation (4) [113]:

$$L_{Dimensions} = \frac{1}{n} \sum (D^* - \bar{D} - \delta)^2 \tag{4}$$

where D^* are the 3D bounding boxes' true dimensions, $\bar{D}$ are the mean dimensions of the objects of a specific class, n is the number of objects in the training batch, and δ is the predicted deviation value with respect to the average that the neural network predicts.

Figure 7 describes the architecture of the Dimensions Estimation Head. The number of fully connected units is (1280, 256, 3) and estimates the object dimensions.

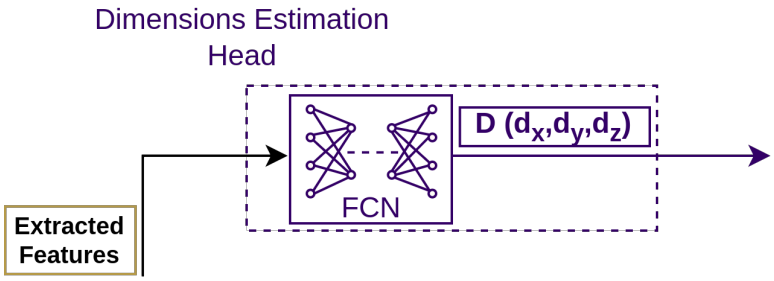

Figure 7. Dimensions Estimation Head.

3.3. Bird's Eye View Bounding Box Center Estimation

The object's visual characteristics and semantic category both influence its Bird's Eye View map (e.g., a truck is longer than a car). Thus, estimating the corresponding Bird's Eye View bounding box center of a detected object can be issued from a deep learning perspective [125].

As shown in Figure 8, the general architecture of the Bird's Eye View Bounding Box Center Estimator is divided into encoder stage and decoder stage. The encoder stage is responsible for encoding the input features into the feature vector. The decoder stage is responsible for decoding the feature vector into the Bird's Eye View centers (t_x,t_z).

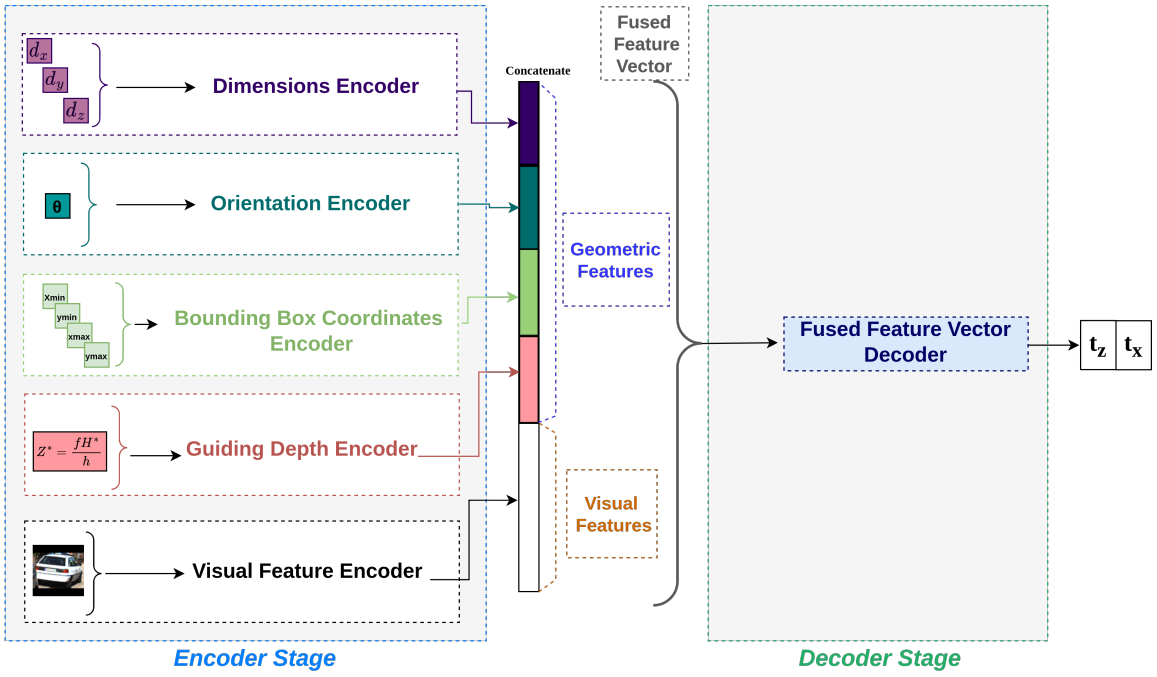

Figure 8. Bird's Eye View center Estimator architecture.

As detailed in Figure 9, the Bird's Eye View Center Estimator encoder stage is composed of five major branches, each branch acts as an encoder to its input feature. The visual feature encoder branch accepts image crops of vehicles detected in the frontal camera view as input, then extracts deep representations using the GhostNet network. This part of the model can extract semantic features from input images despite not knowing where the bounding box is in the scene. It generates a 1280 element feature vector. Each of the remaining four encoder branches generates a 256 element feature vector.

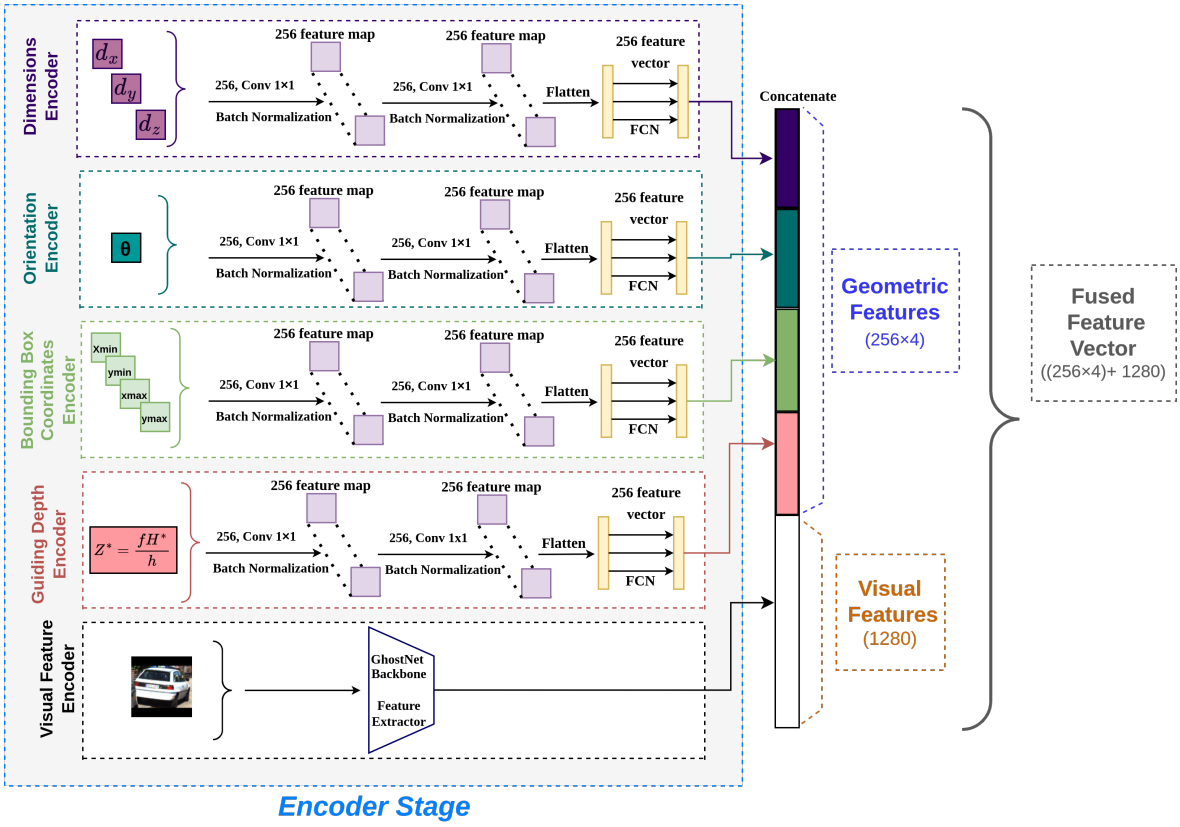

Figure 9. Bird's Eye View Center Estimator encoder stage.

The first branch processes the estimated physical object dimensions, while the second branch takes the estimated object orientation as its input. The third branch accepts the 2D bounding box coordinates from the frontal camera view and the fourth branch takes the ratio between the estimated physical height of the object to the object's visual height in the 2D bounding box. This ratio plays the role of the initial guess of the object depth as shown in Figure 10.

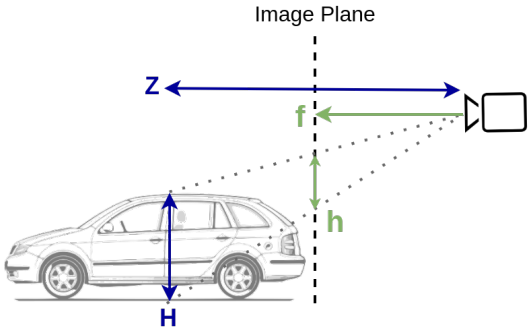

Figure 10. Object physical height to visual height ratio, which acts as a guiding depth estimation.

Depending on the geometry of a pinhole camera, the distance between the object center and the camera (Z) is denoted in accordance with the details presented in Equation (5) [126]

$$Z = \frac{fH}{h} \tag{5}$$

where h is the height of the 2D bounding box, f is the camera focal length, and H is the actual physical height of the object.

As the exact actual physical height (H) is estimated to be (d_y), it can be used to give an initial guide to the depth, as explained in Equation (6) [127]

$$Z^* = \frac{f d_y}{h} \tag{6}$$

where d_y is the estimated physical height and Z^* is the guiding depth.

The four encoded feature vectors in addition to the GhostNet extracted features of Figure 8 are concatenated to generate one geometric-visual fused feature vector of size ($256 \times 4 + 1280 = 2304$) elements and fed to the decoder network which estimates the center coordinate (t_x,t_z) of the Bird's Eye View bounding box, as shown in Figure 11.

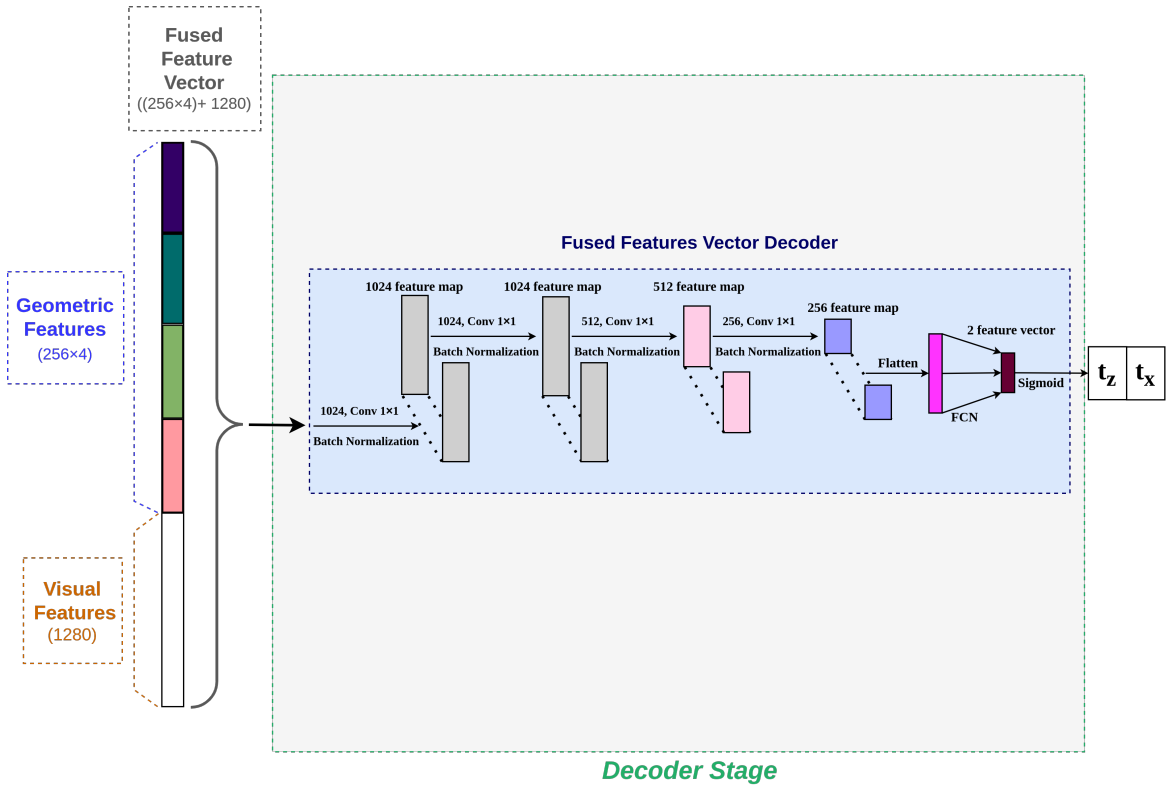

Figure 11. Bird's Eye View Center Estimator decoder stage.

In this way, the above model depicted in Figure 8 accepts the full representative geometric-visual properties of the object and learns to predict the corresponding Bird's Eye View bounding box center of the detected object. The full Bird's Eye View center estimation model was trained on L2 Loss [124] to generate the translation loss ($L_{Translations}$) , as the explanation provided in Equation (7):

$$L_{Translations} = \frac{1}{n} \sum (T^* - T)^2 \tag{7}$$

where T^* are the true Bird's Eye View centers, n is the number of objects in the training batch, and T are the estimated Bird's Eye View centers.

3.4. GhostNet as an Off-the-Shelf Lightweight Feature Extractor

Feature maps are spatial maps created by utilizing a regular convolution layer to inputs. Based on learned convolution filter weights for that layer, these feature maps are responsible for preserving certain feature-rich characterizations of the input image. By examining extracted features in standard convolution layers for evidence that they possess a particular pattern, it is observed that there are multiple close copies of special intrinsic extracted features in the entire set of feature maps developed by the convolution

operation. The generation of which becomes redundant since the convolution operation is quite computationally expensive. Accordingly, the term "Ghost Feature Maps" [121] refers to these redundant duplicates. The actual reason for working on GhostNet is limiting the number of parameters, FLOPs (FLoating point OPerations), and becoming very close to the benchmark accuracy obtained by utilizing the original feature maps. GhostNet generates a percentage of the overall output feature maps, whereas the rest are generated by a simple linear operation. This low-cost linear operation leads to a significant reduction in parameters and FLOPs while maintaining nearly the same performance as the original baseline model [121]. The low-cost linear operation is crafted to be learnable and input-dependent, allowing it to be optimized using backpropagation in the backward pass. The effectiveness of this network as a feature extractor [128–130] is proved to ideally fit the Orientation-Dimensions Estimator, and the Bird's Eye View Center Estimator, which results in overall only 14 million parameters for both models while competing with the state-of-the-art monocular 3D object detection models.

4. Experimental Work

4.1. Implementation Details

As mentioned in the previous section, the proposed MonoGhost network estimates the object's 3D properties from 2D bounding box extracted from monocular images depending on two main stages: The first stage, shown in Figure 5 above, is composed of a feature extractor and three fully connected branches that share the extracted features' pool. This first stage is responsible for estimating the object's 3D dimensions as well as its orientation. The second stage, shown in Figure 8 above, is composed of a feature extractor and four 1D convolution branches, which work as encoders for the supplied visual-geometric features. All the encoded features are fused and fed to a 1D convolution network to work as a decoder that finally predicts the Bird's Eye View center of the object.

4.2. Dataset

The KITTI [22] dataset is a widely available open-source dataset for accurately assessing learning-based methodologies on driving scenes. It is believed to depict accurate representations of autonomous driving scenarios. There are 7481 training images and 7518 testing images in total. Depending on the occlusion, truncation level and the appeared height of the 2D bounding box of object instances, the test metric is divided into easy, moderate, and hard cases [22]. The 3D Object Detection and Bird's Eye View benchmarks are available for testing the proposed method's 3D detection task.

4.2.1. Data Augmentation

Data augmentation techniques were utilized to increase the training dataset, improve the training process, and avoid overfitting. The adopted techniques are random Gauss noise, optical distortion with probability 0.5, and random fog with probability 0.8.

4.2.2. Preprocessing

Applying any complex preprocessing method on the dataset was avoided. Instead, the only preprocessing step was applying resize with padding on the cropped 2D image of the detected object. The aim of this step is to preserve the object properties before being propagated to the Orientation-Dimensions Estimator or to the Bird's Eye View Center Estimator, rather than losing object properties by using scaling only as illustrated by Figure 12. The Orientation-Dimensions Estimator and the Bird's Eye View Center Estimator expect to receive images with fixed dimensions $112 \times 112 \times 3$.

Figure 12. Using resize with padding preserves the object's geometric shape and ratio unlike using resize only.

4.3. Training

The training process was performed for the MonoGhost network on a Geforce RTX 3060 Ti 8G. The two branches of the network were trained independently. The optimizer was chosen to be AdamW [131] with weight decay 1×10^{-3} for both branches.

4.3.1. Orientation-Dimensions Estimator

The Orientation-Dimensions Estimator was trained on batch size of 200 for 250 epochs. The learning rate was set initially at 1×10^{-4}. The scheduler was set to be reduced on plateau by a factor of 0.1, with patience 10, and threshold 1×10^{-4}. The optimizer was chosen to be AdamW with weight decay 1×10^{-3}.

4.3.2. Bird's Eye View Center Estimator

The Bird's Eye View Center Estimator was trained on batch size of 200 for 500 epochs. The learning rate was set initially at 1×10^{-3}. The scheduler was set to cosine annealing with a maximum number of iterations at 4, and a minimum learning rate of 5×10^{-8}.

5. Results and Discussion

As the proposed MonoGhost network is composed of two stages, and each one can be trained independently, the results of the proposed network can be validated on two different benchmarks, the first benchmark (https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=2d, accessed on 5 November 2023) is the object orientation, while the second benchmark (https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=bev, accessed on 5 November 2023) can be used to verify the results of the complete Bird's Eye View estimation. For both benchmarks, an off-the-shelf 2D object detector Faster RCNN [119] is utilized to supply the 2D bounding boxes' coordinates.

The first contribution of MonoGhost network is proposing a deep learning architecture (Bird's Eye View Center Estimator) for estimating Bird's Eye View bounding box of the detected object, depending on encoding the object geometric and visual features using 1D convolution then fusing the encoded features to decode the object Bird's Eye View

translation center. As shown in Figure 13, the average error of Z estimation on the KITTI training dataset was 0.682 m, with standard deviation 0.732 m, and the maximum error was 11.9 m; while the average error of X estimation was 0.243 m, with standard deviation 0.3035 m, and the maximum error was 6.658 m, as stated in Figure 14.

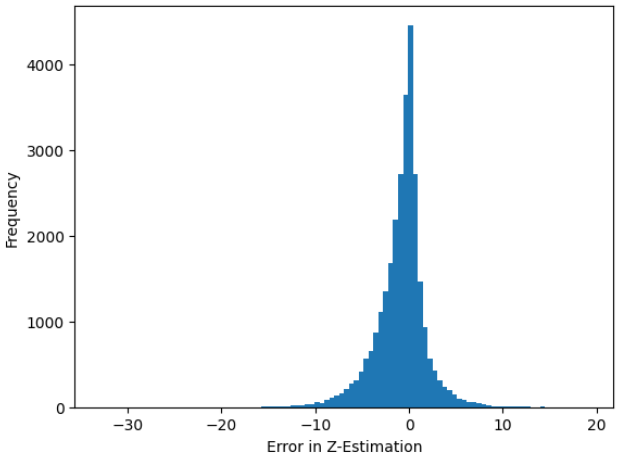

Figure 13. Histogram of Z estimation errors in meters on the KITTI training dataset.

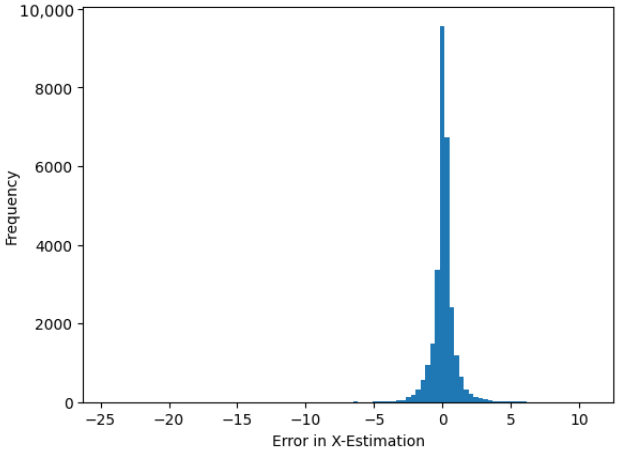

Figure 14. Histogram of X estimation errors in meters on the KITTI training dataset.

Table 1 shows the proposed MonoGhost network results for orientation on KITTI benchmark.

Table 1. MonoGhost network orientation score on KITTI benchmark. An increase in the measured accuracies, as indicated by the ⇑ up arrows, corresponds to better results.

Benchmark	Easy ⇑	Moderate ⇑	Hard ⇑
Car (Detection)	90.79%	83.33%	71.13%
Car (Orientation)	90.23%	82.27%	69.81%

The values presented in Table 1 characterize the accuracy of orientation estimation by the MonoGhost network. It was assessed using the Average Orientation Similarity (AOS) metric, as defined in [22], which incorporates both the average precision (AP) of the 2D detector and the average cosine distance similarity for azimuth orientation.

A sample of the Bird's Eye View results of testing MonoGhost network on KITTI samples is shown in Figure 15.

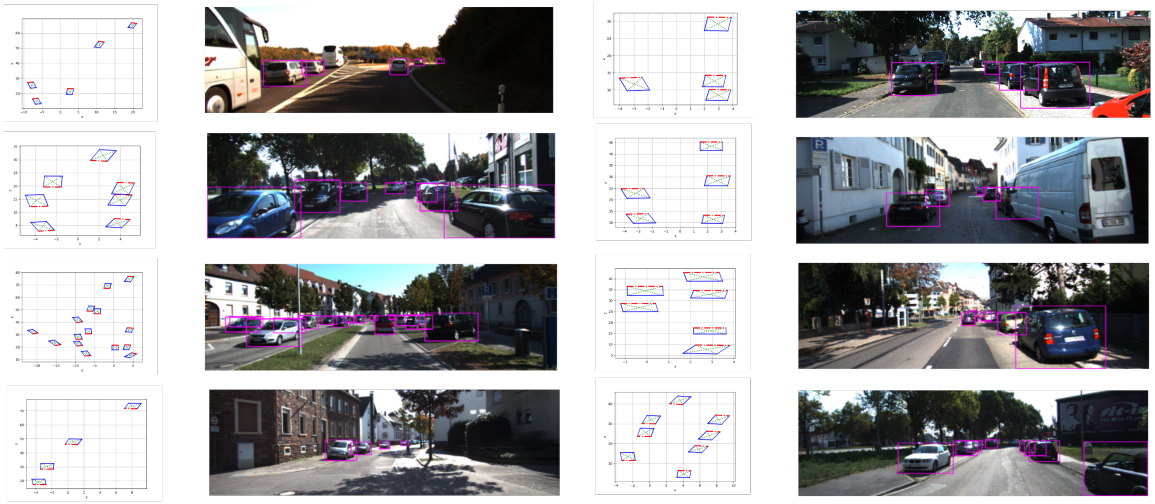

Figure 15. A sample of results inferring the proposed MonoGhost network on the KITTI dataset.

The second contribution of MonoGhost network is preserving the performance of the stable object's depth estimation for KITTI hard object cases without sacrificing the orientation prediction accuracy. Figure 16 shows the average depth estimation error plotted in 10 m intervals. It is shown that the implemented MonoGhost network retains a rather stable error across all the intervals, and scores the best error for ground truth distance further than 60 m. This insight is assured by scoring 15.01% on KITTI hard object cases, which surpasses the performance of the most-known state-of-the-art monocular detection networks. Moreover, Table 1 shows a stable orientation estimation on KITTI's moderate and hard object cases.

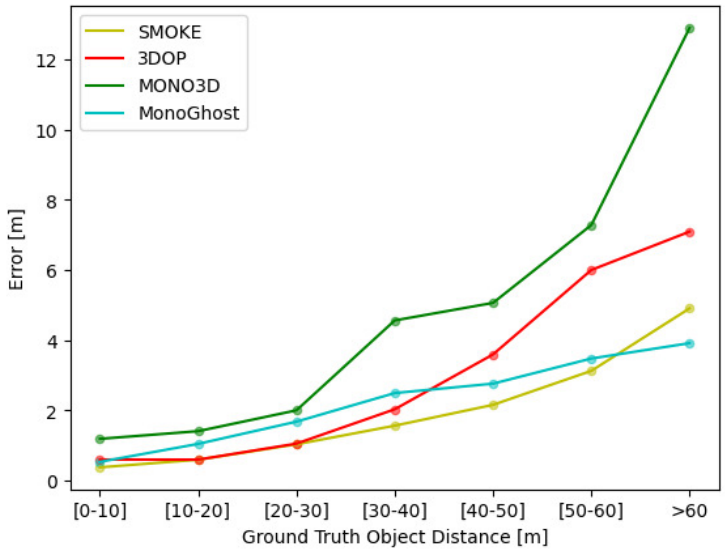

Figure 16. The average depth estimation error plotted in 10 m intervals. The networks used as baselines for comparison are SMOKE [25], 3DOP [96], and MONO3D [132].

The last contributions of MonoGhost network are the simple design of MonoGhost network, which is composed of low computational burden operations and the selection of an efficient lightweight feature extractor. Thus, it is crucial to measure the time performance in order to maintain the real-time requirements. By evaluating the inference time of the model on Geforce RTX 3060 Ti 8G, it shows 0.033 s on average per batch composed of 70 objects; on GeForce GTX 1050 Ti it achieves 0.058 for a batch of 70 objects. The obtained

inference time benchmarks ensure real-time operation even while under-utilizing relatively cheap, not powerful GPUs.

Table 2 presents a comprehensive comparison of the Bird's Eye View results scored between the proposed MonoGhost network and well-known state-of-the-art monocular 3D object detection networks, including PGD-FCOS3D [23], ImVoxelNet [133], SMOKE [25], and M3D-RPN [114], as listed on the KITTI leaderboard. The KITTI Bird's Eye View benchmark employs the PASCAL criteria [134] to evaluate detection performance [22]. The KITTI Bird's Eye View Benchmark defines difficulties based on the following criteria:

- Easy: Minimum bounding box height: 40 pixels, Maximum occlusion level: Fully visible, Maximum truncation: 15%.
- Moderate: Minimum bounding box height: 25 pixels, Maximum occlusion level: Partly occluded, Maximum truncation: 30%.
- Hard: Minimum bounding box height: 25 pixels, Maximum occlusion level: Difficult to see, Maximum truncation: 50%.

These difficulty levels provide a clear framework for assessing the performance of the proposed MonoGhost network in a range of challenging scenarios.

Table 3 presents a detailed analysis of the inference time (time taken to process one frame) and the associated hardware deployment requirements for each method mentioned in Table 2. The table shows that the proposed MonoGhost method has an inference time of 0.03 s on RTX3060TI GPU. This time is shared with three other techniques (PGD-FCOS3D [23], QD-3DT [110], and SMOKE [25]). It is worthwhile to note here that these three methods employed much higher and more expensive GPUs with larger amounts of GPU memories and CUDA cores to achieve the same inference time as the proposed MonoGhost method. It is noteworthy that QD-3DT [110], despite utilizing an RTX2080TI GPU with a relatively lower number of CUDA cores, compensates for this limitation with a significantly higher GPU memory capacity. As a result, it boasts a markedly more powerful hardware configuration compared to the proposed MonoGhost network, which utilizes an RTX3060Ti GPU. This heightened GPU capacity empowers QD-3DT [110] to achieve comparable inference times to the proposed MonoGhost network. Notably, even when QD-3DT [110] employs the more powerful RTX2080TI, its performance ranking remains significantly lower than that of MonoGhost on KITTI Bird's Eye View Leaderboard. This analysis underscores the superior performance of the proposed MonoGhost network, both in terms of inference time and the necessary hardware configuration to achieve efficient inference. The results highlight the MonoGhost network as a compelling choice for real-time applications that can make use of hardware environments with limited resources. It not only excels in terms of inference speed but also minimizes hardware demands, further enhancing its suitability for resource-constrained settings.

By combining the Bird's Eye View results obtained in Table 2 with the inference time comparison from Table 3, the MonoGhost network demonstrates its potential to achieve a stable detection score while maintaining a very reasonable inference time.

Notably, the above results from Monocular 3D object detection methods, in Tables 2 and 3, employ cheap camera sensors and generate depth information by computation or estimation. Their ranks in the KITTI Bird's Eye View benchmark (https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=bev, accessed on 5 November 2023) range from 405 to 424 and their accuracies range from 4% to 26%. This can be achieved without the need for powerful/expensive hardware sensor setup such as the case with LiDAR-based 3D detection methods (e.g., IA-SSD [135], RangeIoUDet [136], StructuralIF [137], and RangeDet [138]). These LiDAR-based 3D detection methods employ comparatively very expensive hardware, require large computational powers, and inherently generate depth perception for the surrounding environment. These methods rank from 68 to 195 in the KITTI Bird's Eye View benchmark (https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=bev, accessed on 5 November 2023) and their accuracies range from 82% to 93%. These two accuracy ranges are completely distinct and can not

be compared together without considering the other underlying hardware/computational factors.

Table 2. A comparison between the Bird's Eye View benchmark scores between the most known state-of-the-art monocular 3D object detection networks and the proposed MonoGhost network as submitted on KITTI car Bird's Eye View benchmark leaderboard (https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=bev, accessed on 5 November 2023). Notably, the rank column represents a global rank, encompassing all detection techniques, including LiDAR-based, stereo-based, depth-aided, and single-image-based methods. The direction of the arrows in each column signifies the trend of increasing model performance.

Method	Rank ⇓	Easy ⇑	Moderate ⇑	Hard ⇑
PGD-FCOS3D [23]	406	**26.89%**	16.51%	13.49%
KM3D-Net [24]	409	23.44%	16.20%	14.47%
D4LCN [101]	410	22.51%	16.02%	12.55%
MonoPair [112]	413	19.28%	14.83%	12.89%
Decoupled-3D [109]	414	23.16%	14.82%	11.25%
QD-3DT [110]	415	20.16%	14.71%	12.76%
SMOKE [25]	416	20.83%	14.49%	12.75%
RTM3D [108]	417	19.17%	14.20%	11.99%
Mono3D_PLiDAR [99]	418	21.27%	13.92%	11.25%
M3D-RPN [114]	420	21.02%	13.67%	10.23%
MonoCInIS [111]	424	22.28%	11.64%	9.95%
MONO3D [132]	-	5.22%	5.19%	4.13%
Proposed MonoGhost network	**405**	24.91%	**16.73%**	**15.01%**

Table 3. A comparison between the most-known state-of-the-art monocular 3D object detection networks and the proposed MonoGhost network depending on the utilized GPU, inference time, and the adopted method. The direction of the arrows in each column signifies the trend of increasing model performance

Method	Time ⇓	Hardware	Adopted Aproach	CUDA Cores ⇓	GPU Memory ⇓
PGD-FCOS3D [23]	0.03 s	4xGTX 1080Ti	Single-Image-Based Approach	4 × 3584	4 × 11 GB
KM3D-Net [24]	0.04 s	1xGTX1080Ti	Single-Image-Based Approach	1 × 3584	1 × 11 GB
D4LCN [101]	0.2 s	4xTesla v100 GPUs	Depth-Aided Approach	1 × 5120	1 × 32 GB
MonoPair [112]	0.057 s	1xGTX1080Ti	Single-Image-Based Approach	1 × 3584	1 × 11 GB
Decoupled-3D [109]	0.08 s	Not mentioned	Single-Image-Based Approach	-	-
QD-3DT [110]	0.03 s	RTX 2080Ti	Single-Image-Based Approach	1 × 4352	1 × 11 GB
SMOKE [25]	0.03 s	4xGeforce TITAN X GPUs	Single-Image-Based Approach	4 × 3072	4 × 12 GB

Table 3. *Cont.*

Method	Time ⇓	Hardware	Adopted Aproach	CUDA Cores ⇓	GPU Memory ⇓
RTM3D [108]	0.05 s	2xGTX1080Ti	Single-Image-Based Approach	2 × 3584	2 × 11 GB
Mono3D_PLiDAR [99]	0.1 s	Not mentioned	Depth-Aided Approach	-	-
M3D-RPN [114]	0.16 s	1xGTX1080Ti	Single-Image-Based Approach	1 × 3584	1 × 11 GB
MonoCInIS [111]	0.13 s	1xGTX1080Ti	Single-Image-Based Approach	1 × 3584	1 × 11 GB
Proposed MonoGhost network	0.03 s	1xRTX3060Ti	Single-Image-Based Approach	1 × 4864	1 × 8 GB

In order to perform a quick ablation study of the Bird's Eye View Bounding Box Center Estimation stage of the proposed MonoGhost network, each of the five inputs to the Bird's Eye View Bounding Box Center Estimation stage were zeroed in turn (one at a time) in order to establish their effect on/contribution to the final output of the system. The results indicated that removing any one of these inputs yielded indefinite NaN (=Not a number) results. This implies that the removal of any of these inputs requires a substantial change in the architecture of the proposed network.

The above results show that the proposed MonoGhost is a highly suitable solution for detection of class "Car" in autonomous driving systems. It is worthwhile to note here that classes other than "Car" have been considered in only five of the techniques provided in Table 2 (PGD-FCOS3D [23], D4LCN [101], MonoPair [112], QD-3DT [110], and M3D-RPN [114]). The proposed MonoGhost network can be employed to detect classes other than cars (e.g., pedestrians, cyclists, etc.); however, this is outside the scope of this paper. Please Note that the KITTI ranking (https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=bev, accessed on 5 November 2023) [22] was conducted based on the moderate score of proposed model. The MonoGhost network proves that the orientation estimation depending on MultiBin discrete–continuous methodology extensively outperforms other methods by utilizing a lightweight feature extractor.

6. Conclusions

This paper presented a novel lightweight architecture to estimate the Bird's Eye View bounding boxes for known object classes from a single image. The method decouples the 3D bounding box estimation problem into two separate tasks, by first estimating the object's geometric features presented by the orientation and dimensions then fusing these estimates with the object visual features in order to output a stable 3D bounding box. The proposed Monocular 3D detection MonoGhost network achieves promising results in terms of accuracy by achieving 24.91% on easy cases, 16.73% on moderate cases, and 15.01% on hard cases of KITTI Bird's Eye View benchmark (https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=bev, accessed on 5 November 2023), with an average inference time of 0.033 s per batch of 70 objects. The proposed MonoGhost network has the potential of integrability with cutting-edge 2D object detection platforms in order to be deployed in autonomous vehicles and in robotic navigation.

Author Contributions: The authors made the following contributions to the research project. A.E.-D. conceptualized and designed the study, including the development of the network architecture. They implemented the code, conducted the experiments, and analyzed the results. Additionally, they conducted a comprehensive literature search and benchmarked the obtained results. A.E.-Z. performed extensive analysis of the dataset and contributed to the analysis and interpretation of the results. They also provided critical input and approvals during the results analysis phase. M.E.-H. contributed to the refinement of the network architecture design and assisted in the literature search. They actively participated in the writing process, including the preparation of the manuscript. Furthermore, they played a key role in the analysis and approval of the research findings. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: In this research, all the used datasets are publicly available for research purposes on the KITTI dataset website (https://www.cvlibs.net/datasets/kitti/index.php). Additionally, all the results obtained from MonoGhost network experiments have been benchmarked against the KITTI benchmark (https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=bev, accessed on 5 November 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Crayton, T.J.; Meier, B.M. Autonomous vehicles: Developing a public health research agenda to frame the future of transportation policy. *J. Transp. Health* **2017**, *6*, 245–252. [CrossRef]
2. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access* **2020**, *8*, 58443–58469. [CrossRef]
3. Shladover, S.E. Review of the state of development of advanced vehicle control systems (AVCS). *Veh. Syst. Dyn.* **1995**, *24*, 551–595. [CrossRef]
4. Vander Werf, J.; Shladover, S.E.; Miller, M.A.; Kourjanskaia, N. Effects of adaptive cruise control systems on highway traffic flow capacity. *Transp. Res. Rec.* **2002**, *1800*, 78–84. [CrossRef]
5. Calvert, S.; Schakel, W.; Van Lint, J. Will automated vehicles negatively impact traffic flow? *J. Adv. Transp.* **2017**, *2017*, 3082781. [CrossRef]
6. Gasser, T.M.; Westhoff, D. BASt-study: Definitions of automation and legal issues in Germany. In Proceedings of the 2012 Road Vehicle Automation Workshop, Irvine, CA, USA, 25 July 2012.
7. International, S. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE Int.* **2018**, *4970*, 1–5.
8. Varotto, S.F.; Hoogendoorn, R.G.; van Arem, B.; Hoogendoorn, S.P. Empirical longitudinal driving behavior in authority transitions between adaptive cruise control and manual driving. *Transp. Res. Rec.* **2015**, *2489*, 105–114. [CrossRef]
9. Nassi, D.; Ben-Netanel, R.; Elovici, Y.; Nassi, B. MobilBye: Attacking ADAS with camera spoofing. *arXiv* **2019**, arXiv:1906.09765.
10. Vivek, K.; Sheta, M.A.; Gumtapure, V. A comparative study of Stanley, LQR and MPC controllers for path tracking application (ADAS/AD). In Proceedings of the 2019 IEEE International Conference on Intelligent Systems and Green Technology (ICISGT), Visakhapatnam, India, 29–30 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 67–674.
11. Gupta, A.; Anpalagan, A.; Guan, L.; Khwaja, A.S. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array* **2021**, *10*, 100057. [CrossRef]
12. Sharma, D. Evaluation and Analysis of Perception Systems for Autonomous Driving. 2020. Available online: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1536525&dswid=-9079 (accessed on 5 November 2023).
13. Liu, Y.; Yixuan, Y.; Liu, M. Ground-aware monocular 3D object detection for autonomous driving. *IEEE Robot. Autom. Lett.* **2021**, *6*, 919–926. [CrossRef]
14. Li, Z.; Du, Y.; Zhu, M.; Zhou, S.; Zhang, L. A survey of 3D object detection algorithms for intelligent vehicles development. *Artif. Life Robot.* **2022**, *27*, 115–122. [CrossRef] [PubMed]
15. Wu, Y.; Wang, Y.; Zhang, S.; Ogai, H. Deep 3D object detection networks using LiDAR data: A review. *IEEE Sens. J.* **2020**, *21*, 1152–1171. [CrossRef]
16. Qian, R.; Lai, X.; Li, X. 3D object detection for autonomous driving: A survey. *Pattern Recognit.* **2022**, *130*, 108796. [CrossRef]
17. Wu, J.; Yin, D.; Chen, J.; Wu, Y.; Si, H.; Lin, K. A survey on monocular 3D object detection algorithms based on deep learning. *J. Phys. Conf. Ser.* **2020**, *1518*, 012049. [CrossRef]
18. Gu, F.; Zhao, H.; Ma, Y.; Bu, P. Camera calibration based on the back projection process. *Meas. Sci. Technol.* **2015**, *26*, 125004. [CrossRef]
19. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.

20. Abhishek, A.V.S.; Kotni, S. Detectron2 Object Detection & Manipulating Images using Cartoonization. *Int. J. Eng. Res. Technol.* **2021**, *10* , 1.
21. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. You only learn one representation: Unified network for multiple tasks. *arXiv* **2021**, arXiv:2105.04206.
22. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conferencef on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 3354–3361.
23. Wang, T.; Zhu, X.; Pang, J.; Lin, D. Probabilistic and Geometric Depth: Detecting Objects in Perspective. In Proceedings of the Conference on Robot Learning (CoRL), London, UK, 8 November 2021.
24. Li, P.; Zhao, H. Monocular 3D detection with geometric constraint embedding and semi-supervised training. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5565–5572. [CrossRef]
25. Liu, Z.; Wu, Z.; Tóth, R. Smoke: Single-stage monocular 3D object detection via keypoint estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 13–19 June 2020; pp. 996–997.
26. Lamdan, Y.; Schwartz, J.T.; Wolfson, H.J. Affine invariant model-based object recognition. *IEEE Trans. Robot. Autom.* **1990**, *6*, 578–589. [CrossRef]
27. Rigoutsos, I.; Hummel, R. Implementation of geometric hashing on the connection machine. In Proceedings of the Workshop on Directions in Automated CAD-Based Vision, Maui, HI, USA, 2–3 June 1991; IEEE Computer Society: Piscataway, NJ, USA, 1991; pp. 76–77.
28. Rigoutsos, I. *Massively Parallel Bayesian Object Recognition*; New York University: New York, NY, USA, 1992.
29. Biegelbauer, G.; Vincze, M.; Wohlkinger, W. Model-based 3D object detection: Efficient approach using superquadrics. *Mach. Vis. Appl.* **2010**, *21*, 497–516. [CrossRef]
30. Bebis, G.; Georgiopoulos, M.; da Vitoria Lobo, N. Learning geometric hashing functions for model-based object recognition. In Proceedings of the IEEE International Conference on Computer Vision, Cambridge, MA, USA, 20–23 June 1995; IEEE: Piscataway, NJ, USA, 1995; pp. 543–548.
31. He, Z.; Feng, W.; Zhao, X.; Lv, Y. 6D pose estimation of objects: Recent technologies and challenges. *Appl. Sci.* **2020**, *11*, 228. [CrossRef]
32. Wang, K.; Xie, J.; Zhang, G.; Liu, L.; Yang, J. Sequential 3D human pose and shape estimation from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 7275–7284.
33. Li, X.; Wang, H.; Yi, L.; Guibas, L.J.; Abbott, A.L.; Song, S. Category-level articulated object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 April 2020; pp. 3706–3715.
34. Zhang, Z.; Hu, L.; Deng, X.; Xia, S. Weakly supervised adversarial learning for 3D human pose estimation from point clouds. *IEEE Trans. Vis. Comput. Graph.* **2020**, *26*, 1851–1859. [CrossRef] [PubMed]
35. Guo, Z.; Chai, Z.; Liu, C.; Xiong, Z. A fast global method combined with local features for 6d object pose estimation. In Proceedings of the 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Hong Kong, China, 8–12 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
36. Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image Vis. Comput.* **1992**, *10*, 145–155. [CrossRef]
37. Yu, H.; Fu, Q.; Yang, Z.; Tan, L.; Sun, W.; Sun, M. Robust robot pose estimation for challenging scenes with an RGB-D camera. *IEEE Sens. J.* **2018**, *19*, 2217–2229. [CrossRef]
38. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 2564–2571.
39. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. Brief: Binary robust independent elementary features. In Proceedings of the Computer Vision—ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; Proceedings, Part IV 11; Springer: Berlin/Heidelberg, Germany, 2010; pp. 778–792.
40. Nospes, D.; Safronov, K.; Gillet, S.; Brillowski, K.; Zimmermann, U.E. Recognition and 6D pose estimation of large-scale objects using 3D semi-global descriptors. In Proceedings of the 2019 16th International Conference on Machine Vision Applications (MVA), Tokyo, Japan, 27–31 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
41. Park, K.; Patten, T.; Prankl, J.; Vincze, M. Multi-task template matching for object detection, segmentation and pose estimation using depth images. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 7207–7213.
42. Konishi, Y.; Hattori, K.; Hashimoto, M. Real-time 6D object pose estimation on CPU. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 3451–3458.
43. Tamadazte, B.; Marchand, E.; Dembélé, S.; Le Fort-Piat, N. CAD model-based tracking and 3D visual-based control for MEMS microassembly. *Int. J. Robot. Res.* **2010**, *29*, 1416–1434. [CrossRef]
44. Brachmann, E.; Michel, F.; Krull, A.; Yang, M.Y.; Gumhold, S.; Rother, C. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In Proceedings of the IEEE Conferencef on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3364–3372.

45. Marullo, G.; Tanzi, L.; Piazzolla, P.; Vezzetti, E. 6D object position estimation from 2D images: A literature review. *Multimed. Tools Appl.* **2022**, *82*, 24605–24643. [CrossRef]
46. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
47. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
48. Miyake, E.; Takubo, T.; Ueno, A. 3D Pose Estimation for the Object with Knowing Color Symbol by Using Correspondence Grouping Algorithm. In Proceedings of the 2020 IEEE/SICE International Symposium on System Integration (SII), Honolulu, HI, USA, 12–15 January 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 960–965.
49. Zhang, X.; Jiang, Z.; Zhang, H.; Wei, Q. Vision-Based Pose Estimation for Textureless Space Objects by Contour Points Matching. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 2342–2355. [CrossRef]
50. Konishi, Y.; Hanzawa, Y.; Kawade, M.; Hashimoto, M. Fast 6D pose estimation from a monocular image using hierarchical pose trees. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 398–413.
51. Lim, J.J.; Khosla, A.; Torralba, A. Fpm: Fine pose parts-based model with 3D cad models. In Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part VI 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 478–493.
52. Muñoz, E.; Konishi, Y.; Murino, V.; Del Bue, A. Fast 6D pose estimation for texture-less objects from a single RGB image. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 5623–5630. [CrossRef]
53. Peng, J.; Xu, W.; Liang, B.; Wu, A.G. Virtual stereovision pose measurement of noncooperative space targets for a dual-arm space robot. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 76–88. [CrossRef]
54. Chaumette, F.; Hutchinson, S. Visual servo control. II. Advanced approaches [Tutorial]. *IEEE Robot. Autom. Mag.* **2007**, *14*, 109–118. [CrossRef]
55. Wnuk, M.; Pott, A.; Xu, W.; Lechler, A.; Verl, A. Concept for a simulation-based approach towards automated handling of deformable objects—A bin picking scenario. In Proceedings of the 2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Auckland, New Zealand, 21–23 November 2017; pp. 1–6. [CrossRef]
56. Kratochvil, B.E.; Dong, L.; Nelson, B.J. Real-time rigid-body visual tracking in a scanning electron microscope. *Int. J. Robot. Res.* **2009**, *28*, 498–511. [CrossRef]
57. Guo, J.; Wu, P.; Wang, W. A precision pose measurement technique based on multi-cooperative logo. *J. Phys. Conf. Ser.* **2020**, *1607*, 012047. [CrossRef]
58. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; IEEE: Piscataway, NJ, USA, 2005; Volume 1, pp. 886–893.
59. Hinterstoisser, S.; Cagniart, C.; Ilic, S.; Sturm, P.; Navab, N.; Fua, P.; Lepetit, V. Gradient response maps for real-time detection of textureless objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 876–888. [CrossRef]
60. Solina, F.; Bajcsy, R. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 131–147. [CrossRef]
61. Roomi, M.; Beham, D. A Review Of Face Recognition Methods. *Int. J. Pattern Recognit. Artif. Intell.* **2013**, *27*, 1356005. [CrossRef]
62. Vishwakarma, V.P.; Pandey, S.; Gupta, M. An illumination invariant accurate face recognition with down scaling of DCT coefficients. *J. Comput. Inf. Technol.* **2010**, *18*, 53–67. [CrossRef]
63. Muñoz, E.; Konishi, Y.; Beltran, C.; Murino, V.; Del Bue, A. Fast 6D pose from a single RGB image using Cascaded Forests Templates. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 4062–4069.
64. Salganicoff, M.; Ungar, L.H.; Bajcsy, R. Active learning for vision-based robot grasping. *Mach. Learn.* **1996**, *23*, 251–278. [CrossRef]
65. Chevalier, L.; Jaillet, F.; Baskurt, A. Segmentation and Superquadric Modeling of 3D Objects. February 2003. Available online: http://wscg.zcu.cz/wscg2003/Papers_2003/D71.pdf (accessed on 5 November 2023).
66. Vilar, C.; Krug, S.; O'Nils, M. Realworld 3D object recognition using a 3D extension of the hog descriptor and a depth camera. *Sensors* **2021**, *21*, 910. [CrossRef] [PubMed]
67. O'Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Hernandez, G.V.; Krpalkova, L.; Riordan, D.; Walsh, J. Deep learning vs. traditional computer vision. In *Proceedings of the Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC)*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11, pp. 128–144.
68. Li, J.; Sun, Y.; Luo, S.; Zhu, Z.; Dai, H.; Krylov, A.S.; Ding, Y.; Shao, L. P2V-RCNN: Point to voxel feature learning for 3D object detection from point clouds. *IEEE Access* **2021**, *9*, 98249–98260. [CrossRef]
69. Li, J.; Luo, S.; Zhu, Z.; Dai, H.; Krylov, A.S.; Ding, Y.; Shao, L. 3D IoU-Net: IoU guided 3D object detector for point clouds. *arXiv* **2020**, arXiv:2004.04962.
70. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. Pv-rcnn: Point-voxel feature set abstraction for 3D object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10529–10538.

71. Shi, S.; Wang, X.; Li, H. Pointrcnn: 3D object proposal generation and detection from point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 770–779.
72. Mao, J.; Shi, S.; Wang, X.; Li, H. 3D object detection for autonomous driving: A review and new outlooks. *arXiv* **2022**, arXiv:2206.09474.
73. Fernandes, D.; Silva, A.; Névoa, R.; Simões, C.; Gonzalez, D.; Guevara, M.; Novais, P.; Monteiro, J.; Melo-Pinto, P. Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy. *Inf. Fusion* **2021**, *68*, 161–191. [CrossRef]
74. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3D object detection. In Proceedings of the IEEE Conferencef on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499.
75. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [CrossRef] [PubMed]
76. Zhu, B.; Jiang, Z.; Zhou, X.; Li, Z.; Yu, G. Class-balanced grouping and sampling for point cloud 3D object detection. *arXiv* **2019**, arXiv:1908.09492.
77. Ge, R.; Ding, Z.; Hu, Y.; Wang, Y.; Chen, S.; Huang, L.; Li, Y. Afdet: Anchor free one stage 3D object detection. *arXiv* **2020**, arXiv:2006.12671.
78. Yin, T.; Zhou, X.; Krahenbuhl, P. Center-based 3D object detection and tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 19–25 June 2021; pp. 11784–11793.
79. Li, F.; Jin, W.; Fan, C.; Zou, L.; Chen, Q.; Li, X.; Jiang, H.; Liu, Y. PSANet: Pyramid splitting and aggregation network for 3D object detection in point cloud. *Sensors* **2020**, *21*, 136. [CrossRef]
80. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the IEEE Conferencef on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
81. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30* , 1, 5, 7.
82. Zarzar, J.; Giancola, S.; Ghanem, B. PointRGCN: Graph convolution networks for 3D vehicles detection refinement. *arXiv* **2019**, arXiv:1911.12236.
83. Shin, K.; Kwon, Y.P.; Tomizuka, M. Roarnet: A robust 3D object detection based on region approximation refinement. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Dearborn, MI, USA, 9–12 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 2510–2515.
84. Meyer, G.P.; Laddha, A.; Kee, E.; Vallespi-Gonzalez, C.; Wellington, C.K. Lasernet: An efficient probabilistic 3D object detector for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12677–12686.
85. Vora, S.; Lang, A.H.; Helou, B.; Beijbom, O. Pointpainting: Sequential fusion for 3D object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4604–4612.
86. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705.
87. Zhao, X.; Liu, Z.; Hu, R.; Huang, K. 3D object detection using scale invariant and feature reweighting networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Hawaii, HI, USA, 27 January–1 February 2019; Volume 33, pp. 9267–9274.
88. Wang, Z.; Jia, K. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1742–1749.
89. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum pointnets for 3D object detection from rgb-d data. In Proceedings of the IEEE Conferencef on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 918–927.
90. Rahman, M.M.; Tan, Y.; Xue, J.; Lu, K. Notice of violation of IEEE publication principles: Recent advances in 3D object detection in the era of deep neural networks: A survey. *IEEE Trans. Image Process.* **2019**, *29*, 2947–2962. [CrossRef]
91. Chang, J.R.; Chen, Y.S. Pyramid stereo matching network. In Proceedings of the IEEE Conferencef on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5410–5418.
92. Zhang, F.; Prisacariu, V.; Yang, R.; Torr, P.H. Ga-net: Guided aggregation net for end-to-end stereo matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 185–194.
93. Wang, Y.; Lai, Z.; Huang, G.; Wang, B.H.; Van Der Maaten, L.; Campbell, M.; Weinberger, K.Q. Anytime stereo image depth estimation on mobile devices. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 5893–5900.
94. Guo, X.; Yang, K.; Yang, W.; Wang, X.; Li, H. Group-wise correlation stereo network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3273–3282.
95. Kendall, A.; Martirosyan, H.; Dasgupta, S.; Henry, P.; Kennedy, R.; Bachrach, A.; Bry, A. End-to-end learning of geometry and context for deep stereo regression. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 66–75.
96. Chen, X.; Kundu, K.; Zhu, Y.; Berneshawi, A.G.; Ma, H.; Fidler, S.; Urtasun, R. 3D object proposals for accurate object class detection. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1.
97. Kar, A.; Häne, C.; Malik, J. Learning a multi-view stereo machine. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 2.

98. Ma, X.; Wang, Z.; Li, H.; Zhang, P.; Ouyang, W.; Fan, X. Accurate monocular 3D object detection via color-embedded 3D reconstruction for autonomous driving. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6851–6860.
99. Weng, X.; Kitani, K. Monocular 3D object detection with pseudo-lidar point cloud. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3, 6, 11.
100. Wang, Y.; Chao, W.L.; Garg, D.; Hariharan, B.; Campbell, M.; Weinberger, K.Q. Pseudo-lidar from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 8445–8453.
101. Ding, M.; Huo, Y.; Yi, H.; Wang, Z.; Shi, J.; Lu, Z.; Luo, P. Learning depth-guided convolutions for monocular 3D object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 1000–1001.
102. Wang, L.; Du, L.; Ye, X.; Fu, Y.; Guo, G.; Xue, X.; Feng, J.; Zhang, L. Depth-conditioned dynamic message propagation for monocular 3D object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 454–463.
103. Huang, K.C.; Wu, T.H.; Su, H.T.; Hsu, W.H. Monodtr: Monocular 3D object detection with depth-aware transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4012–4021.
104. Simonelli, A.; Bulo, S.R.; Porzi, L.; Ricci, E.; Kontschieder, P. Towards generalization across depth for monocular 3D object detection. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXII 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 767–782.
105. Simonelli, A.; Bulo, S.R.; Porzi, L.; López-Antequera, M.; Kontschieder, P. Disentangling monocular 3D object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1991–1999.
106. Ma, X.; Zhang, Y.; Xu, D.; Zhou, D.; Yi, S.; Li, H.; Ouyang, W. Delving into localization errors for monocular 3D object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4721–4730.
107. Zhang, Y.; Lu, J.; Zhou, J. Objects are different: Flexible monocular 3D object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3289–3298.
108. Li, P.; Zhao, H.; Liu, P.; Cao, F. Rtm3d: Real-time monocular 3D detection from object keypoints for autonomous driving. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part III 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 644–660.
109. Cai, Y.; Li, B.; Jiao, Z.; Li, H.; Zeng, X.; Wang, X. Monocular 3D object detection with decoupled structured polygon estimation and height-guided depth estimation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 10478–10485.
110. Hu, H.N.; Yang, Y.H.; Fischer, T.; Darrell, T.; Yu, F.; Sun, M. Monocular quasi-dense 3D object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 1992–2008. [CrossRef]
111. Heylen, J.; De Wolf, M.; Dawagne, B.; Proesmans, M.; Van Gool, L.; Abbeloos, W.; Abdelkawy, H.; Reino, D.O. Monocinis: Camera independent monocular 3D object detection using instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 20–25 June 2021; pp. 923–934.
112. Chen, Y.; Tai, L.; Sun, K.; Li, M. Monopair: Monocular 3D object detection using pairwise spatial relationships. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12093–12102.
113. Mousavian, A.; Anguelov, D.; Flynn, J.; Kosecka, J. 3D bounding box estimation using deep learning and geometry. In Proceedings of the IEEE Conferencef on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7074–7082.
114. Brazil, G.; Liu, X. M3d-rpn: Monocular 3D region proposal network for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9287–9296.
115. Li, B.; Ouyang, W.; Sheng, L.; Zeng, X.; Wang, X. Gs3d: An efficient 3D object detection framework for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1019–1028.
116. Xiang, Y.; Schmidt, T.; Narayanan, V.; Fox, D. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv* **2017**, arXiv:1711.00199.
117. Lu, Y.; Ma, X.; Yang, L.; Zhang, T.; Liu, Y.; Chu, Q.; Yan, J.; Ouyang, W. Geometry uncertainty projection network for monocular 3D object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual Event, 11–17 October 2021; pp. 3111–3121.
118. Huang, S.; Chen, Y.; Yuan, T.; Qi, S.; Zhu, Y.; Zhu, S.C. Perspectivenet: 3D object detection from a single rgb image via perspective points. *Adv. Neural Inf. Process. Syst.* **2019**, *32* , 8.
119. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
120. Daniilidis, K.; Klette, R. *Imaging Beyond the Pinhole Camera*; Springer: Berlin/Heidelberg, Germany, 2006.

121. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
122. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
123. Kingsbury, B. Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling. In Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 3761–3764.
124. Bühlmann, P.; Yu, B. Boosting with the L 2 loss: Regression and classification. *J. Am. Stat. Assoc.* **2003**, *98*, 324–339. [CrossRef]
125. Palazzi, A.; Borghi, G.; Abati, D.; Calderara, S.; Cucchiara, R. Learning to map vehicles into bird's eye view. In Proceedings of the Image Analysis and Processing—ICIAP 2017: 19th International Conference, Catania, Italy, 11–15 September 2017; Proceedings, Part I 19; Springer: Berlin/Heidelberg, Germany, 2017; pp. 233–243.
126. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003.
127. Shi, X.; Ye, Q.; Chen, X.; Chen, C.; Chen, Z.; Kim, T.K. Geometry-based distance decomposition for monocular 3D object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 15172–15181.
128. Liu, T.; Zhou, B.; Zhao, Y.; Yan, S. Ship detection algorithm based on improved YOLO V5. In Proceedings of the 2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE), Dalian, China, 15–17 July 2021; IEEE: Dalian, China, 2021; pp. 483–487.
129. Zhou, D.; Wang, B.; Zhu, C.; Zhou, F.; Wu, H. A light-weight feature extractor for lithium-ion battery health prognosis. *Reliab. Eng. Syst. Saf.* **2023**, *237*, 109352. [CrossRef]
130. Chi, J.; Guo, S.; Zhang, H.; Shan, Y. L-GhostNet: Extract Better Quality Features. *IEEE Access* **2023**, *11*, 2361–2374. [CrossRef]
131. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
132. Chen, X.; Kundu, K.; Zhang, Z.; Ma, H.; Fidler, S.; Urtasun, R. Monocular 3D object detection for autonomous driving. In Proceedings of the IEEE Conferencef on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2147–2156.
133. Rukhovich, D.; Vorontsova, A.; Konushin, A. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3D object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI,USA, 4–8 January 2022; pp. 2397–2406.
134. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
135. Zhang, Y.; Hu, Q.; Xu, G.; Ma, Y.; Wan, J.; Guo, Y. Not all points are equal: Learning highly efficient point-based detectors for 3D lidar point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 18953–18962.
136. Liang, Z.; Zhang, Z.; Zhang, M.; Zhao, X.; Pu, S. Rangeioudet: Range image based real-time 3D object detector optimized by intersection over union. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7140–7149.
137. An, P.; Liang, J.; Yu, K.; Fang, B.; Ma, J. Deep structural information fusion for 3D object detection on LiDAR–camera system. *Comput. Vis. Image Underst.* **2022**, *214*, 103295. [CrossRef]
138. Fan, L.; Xiong, X.; Wang, F.; Wang, N.; Zhang, Z. Rangedet: In defense of range view for lidar-based 3D object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2918–2927.