Surround-View Vision-based 3D Detection for Autonomous Driving: A Survey

Apoorv Singh
Motional
Carnegie Mellon University
apoorv.singh@motional.com

Abstract

Vision-based 3D Detection task is a fundamental task for the perception of an autonomous driving system, which has piqued interest amongst many researchers and autonomous driving engineers. However, achieving a rather good 3D BEV (Bird's Eye View) performance is not an easy task using 2D sensor input data of monocular cameras. This paper provides a literature survey of the existing Vision-Based 3D detection methods focused on autonomous driving. We have made detailed analyses of over 60 papers leveraging Vision BEV detection approaches and binned them into different sub-groups for an easier understanding of the common trends. Moreover, we have highlighted how the literature and industry trends have moved towards surround-view image-based methods and noted thoughts on what special cases these surround-view methods address. In conclusion, we provoke thoughts of 3D Vision techniques for future research based on the shortcomings of the current methods, including the direction of collaborative perception.

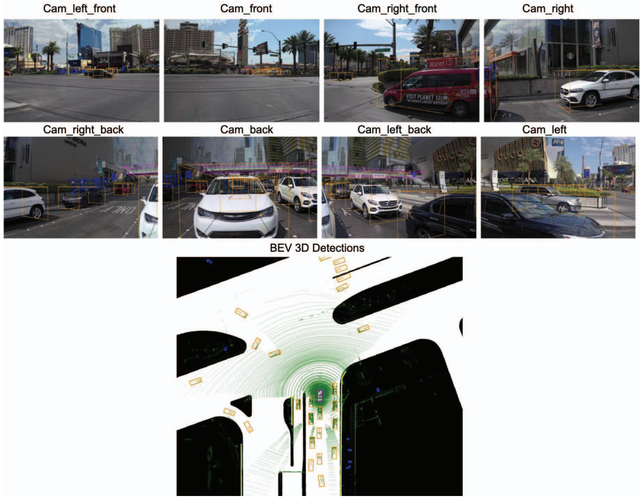

Figure 1. Surround-view Image 3D Detector in autonomous driving. Ground-truth 3D boxes overlaid over surround-images in perspective view (top); Ground-truth 3D boxes overlaid over BEV HD Map (bottom), with ego car in pink.

1. Introduction

Object detection is a trivial task for humans. Pretty much any teenager can look at the scene from the car's windscreen and place all the agents, dynamic or static, in a mental BEV (Bird's Eye View) map. This virtual map may include per-agent information as, but not limited to, center coordinates, dimensions, orientation angle, etc. However, teaching this to a computer has been a nearly impossible task until the turn of the last decade. This task entails identifying and localizing an object's instances (like cars, humans, street signs, etc.) within the field of view as shown in 1. Similarly, classification, segmentation, dense-depth estimation, motion prediction, scene understanding, etc., are other fundamental problems in computer vision.

Early object detection models were built on hand-crafted feature extractors such as Viola-Jones detector [34], Histogram of Oriented Gradients (HOG) [7] etc. These were SoTA (State-of-the-art) of their time; however, compared to the current methods, they are slow, inaccurate, and not scalable on generic datasets. The introduction of convolutional neural networks (CNNs) and deep learning for image classification changed the landscape of visual perception. CNN's use in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 challenge by AlexNet [18] has inspired further research on CNNs in the computer vision industry. Mainstream applications for 3D object detection lie around autonomous driving, mobile-robotic vision, security cameras, etc. Limited Field-of-view (FOV) of cameras has led researchers to the next breakthrough problem-statement of *how to leverage views from multiple cameras to reason the* 360° *surroundings.*

This survey on Surround-view Vision-based 3D object detection provides a comprehensive review of deep learning-based methods and architectures in the recent past.

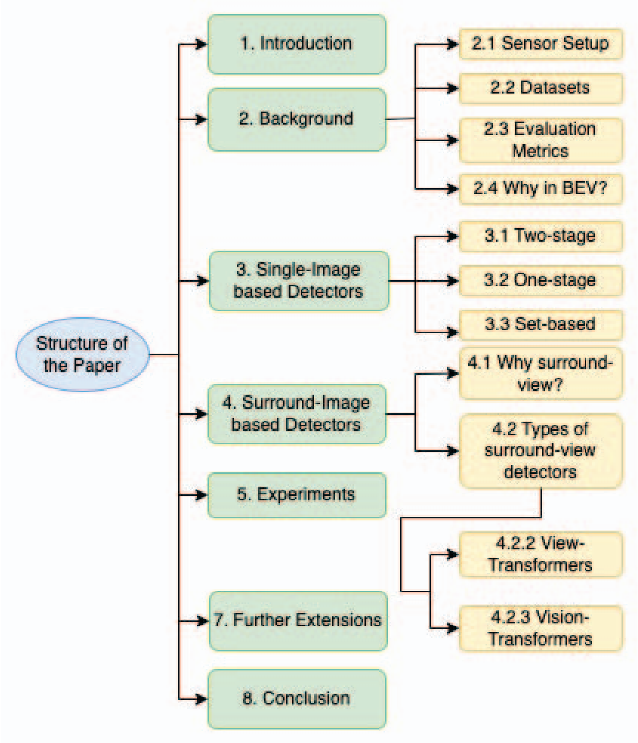

Figure 2. Structure of this Survey Paper.

The main contributions of this paper are as follows:

- This paper provides an in-depth analysis of major single-view detector baselines that inspired surround-view detector research in 3D object detection tasks using cameras.
- This paper provides further analysis of major surround-view detector trends currently in development in the computer-vision community, categorizing them for readers to follow through easily.
- This paper provided detailed background on evaluation metrics and datasets used to evaluate and compare the above methods.
- This paper makes a detailed analysis of the remaining problems. It introduces several potential research directions about the BEV 3D image object detectors, opening a possible door for future research.

The rest of the paper is organized as follows: We first look at the background information required to understand autonomous driving 3D detections viz., evaluation metrics, datasets, annotations, etc. in 2. Then, we introduce single-image-based detection methods and SoTA approaches that inspired surround-view detectors approaches in 3. In 4, we dive into details for surround-view-based detection approaches focused on autonomous driving. We then report and analyze the performance of these approaches on our previously defined metrics in 5. Then in 6, we report possible research extensions on surround-view object detection methods that may enlighten future research. Finally, in 7, we conclude the paper.

2. Background

To cover the basics required to understand 3D BEV object detection tasks, we discuss four aspects: Sensor setup on an autonomous vehicle (AV); frequently used datasets; common evaluation metrics for detection tasks in autonomous driving, and Why Bird's Eye View (BEV) is important for an AV camera perception?

2.1. Sensor Setup

Before we even look at how cameras are set up in an autonomous vehicle (AV), let's try to understand why we need cameras in the first place. Cameras have the most densely packed information compared to other sensors, making them one of the most challenging sensors to extract information from in an AV but the most useful simultaneously. To understand this mathematically, let us first look at the number of data points in each visualization shown in 3. Take these data points (floating point numbers) as the input to the perception algorithm for a sensor to cover 360° view that is responsible for making decisions for an AV.

Let's start with a multi-camera:

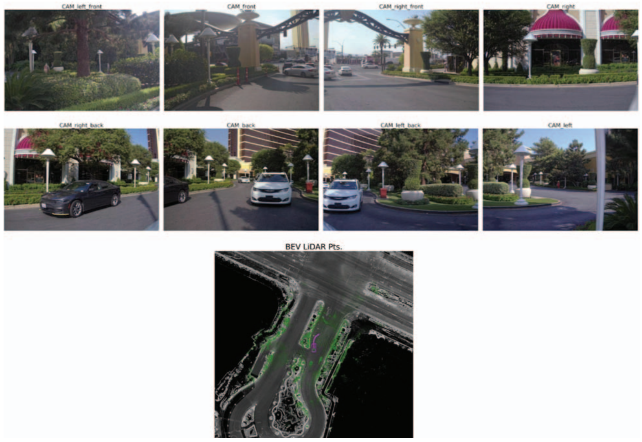

Figure 3. Surround-view 8 camera images (top); LiDAR Point Cloud overlayed over an HD Map (bottom). Key: Green points: LiDAR point cloud; Pink box: Ego Autonomous vehicle; Gray map: Pre-computed HD map with color intensity.

Number of cameras: 8;
The number of pixels per camera: $2,000 * 3,000$ (image pixel resolution: width*height);

Representation of a pixel: 3 (three channeled RGB value). *This brings total parameters to* $8*2000*3000*3 = 144M$ *float numbers!*
Similar comparison with a LiDAR now:
Number of LiDAR points in a point cloud: 250,000;
Representation of each LiDAR point: 4 (3D coordinate i.e. *x, y, z* and reflectance).
This brings total parameters to: $250,000 * 4 = 1M$ *float numbers!*

These numbers and visualizations as in 3 should be enough to prove our point of *"the key role cameras play in the AV perception to perceive the environment."*

A camera is one of the least expensive sensors, unlike other laser-based sensors. However, cameras are spectacularly better for detecting long-range objects and extracting vision-based road cues like the state of traffic lights, stop signs, etc., compared to any other laser-based sensor. Setup of surround-view cameras on an AV may vary depending on different autonomous car manufacturers, but typically there are $6 \sim 12$ cameras per vehicle. These many cameras are needed to cover the entire surrounding 3D scene. We are limited to using cameras with normal FOV (Field of view); otherwise, we may get image distortions beyond recovery, like with Fish-eye cameras (Wide FOV), which are only good for up to a few tens of meters. A perception sensor set up in one of the most cited benchmark-dataset, nuScenes [1] in the AV space can be seen in 4.

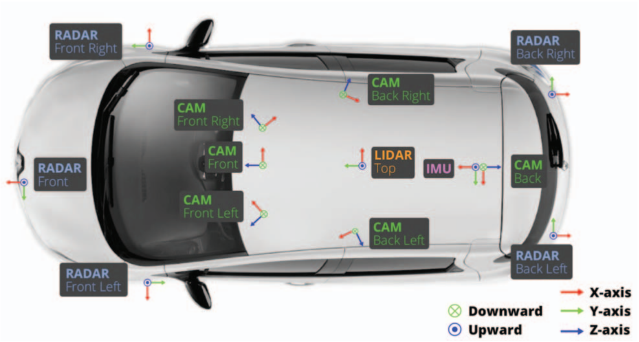

Figure 4. Sensor setup for an Autonomous vehicle in nuScenes [1] benchmark dataset.

2.2. Datasets

nuScenes[1], KITTI[8], Waymo Open Dataset (WOD)[31] are the three most commonly used datasets for 3D BEV object detection task. Apart from them, H3D[28], Lyft L5[13], and Argoverse[3] can also be used for BEV perception tasks. nuSences contains 1000 scenes with a duration of 20 seconds each. They contain six calibrated images covering the 360° view of the road. Sensor setup with nuScenes can be seen in 4. KITTI was the seminal work on the autonomous driving dataset. It consists of a smaller sample of data than the more recent ones. Waymo Open Dataset (WOD) is another large-scale autonomous driving dataset with 798 training sequences, 202 validation, and 150 testing sequences, respectively. Argoverse 2 also contains 1000 scenes with LiDARs, stereo-imagery, ring-camera imagery, a.k.a surround-cameras imagery. Detailed information on this dataset is given as in 1.

2.3. Evaluation Metrics

3D object detectors use multiple criteria to measure the performance of the detectors, viz., precision and recall. However, mean Average Precision (mAP) is the most common evaluation metric. Intersection over Union (IoU) is the ratio of the area of overlap and the area of the union between the predicted box and ground-truth box. An IoU threshold value (generally 0.5) is used to judge if a prediction box matches any particular ground truth box. If IoU is greater than the threshold, then that prediction is treated as a True Positive (TP); else, it is a False Positive (FP). A ground-truth object which fails to detect with any prediction box is treated as a False Negative (FN). Precision is the fraction of relevant instances among the retrieved instances, while recall is the fraction of relevant instances that were retrieved.

$$Precision = TP/(TP + FP) \tag{1}$$

$$Recall = TP/(TP + FN) \tag{2}$$

Based on the above equations, average precision is computed separately for each class. To compare the performance between different detectors (mAP) is used. It is a weighted mean based on the number of ground truths per class. Alternatively, the F1 score is the second most common detection metric, the weighted average of precision and recall. Higher *AP* detectors perform better when the model is deployed at varied confidence thresholds. However, a higher *max-F1* score detector is used when the model is to be deployed at a known fixed optimal-confidence threshold score.

$$F1 = 2 * Precision * Recall/(Precision + Recall) \tag{3}$$

In addition, there are a few dataset-specific metrics viz., KITTI introduces Average Orientation Similarity (AOS), which evaluates the quality of orientation estimation of boxes on the ground plane. mAP metric only considers the 3D position of the objects. However, it ignores the effects of both dimension and orientation. About that, nuScenes introduces TP metrics viz., Average Translation Error (ATE), Average Scale Error (ASE), and Average Orientation Error (AOE). WOD introduces

Table 1. Information on benchmark dataset commonly used for 3D BEV Object Detection using cameras in autonomous driving.

Dataset	Cameras	Scenes	Train	Test	Boxes	Classes	Temporal	LiDAR	RADAR
nuScenes	6	1,000	28,130	6,008	1.4M	10	√	√	√
KITTI (3D)	-	-	7,418	7,518	200K	3	√	√	×
WOD	5	1,150	122,200	40,077	12M	4	√	√	×
Argoverse	7	113	39,384	12,507	993K	15	√	√	×
Lyft L5	6	366	22,690	27,468	1.3M	9	√	√	×
H3D	3	160	8,873	13,678	1.1M	8	√	√	×

Average Precision weighted by Heading (APH) as its main metric. It takes heading/ orientation information into account as well. Also, given depth confusion for 2D sensors like cameras, WOD introduces Longitudinal Error Tolerant 3D Average Precision(LET-3D-AP), which emphasizes more on lateral errors more than longitudinal errors in predictions.

2.4. Why BEV (Bird's Eye View)?

There are several reasons why using the 3D agent's representation in the Bird's Eye View makes more practical sense for autonomous driving:

- It makes fusion with the other 360° sensors, i.e., LiDARs and RADARs, more natural as these laser-based sensors operate in the BEV space natively.

- If we operate in BEV, we can model the temporal consistency of the dynamic scene much better. Motion compensation, i.e., translation and rotation modeling in BEV agents, is much more trivial than the perspective view (camera view). For example, In BEV view: Pose change depends just on the motion of the agent, whereas in perspective-view, pose change depends on the depth and motion of the agent.

- Scale of the objects are consistent in BEV space but not so much in the perspective view. In perspective, view objects appear bigger when closer to the viewpoint. Hence, BEV space makes it easier to learn range-agnostic scale features.

- In autonomous driving, downstream tasks after perception, like motion prediction and motion planning, operate on the BEV space natively. It makes natural sense for all the software stacks to work in a common coordinate-view system on a robotic platform.

- Newly researched field, Collaborative perception, which we will discuss in Section 6, also utilizes BEV representation to represent all the agents at a common coordinate system.

3. Single-Image Based Detectors

We have divided single-view image-based object detection methods into two-stage, single-stage, and set-based detectors. However, we would like to mention pioneer works like Viola-Jones [34], HOG Detector [7], Deformable Parts Model (DPM) [10], which have revolutionized computer vision with PASCAL VOC challenge in 2009 [44]. These approaches use classical computer-vision techniques that extract human-designed heuristic features.

3.1. Two-stage Detectors

This is a class of detectors divided into two stages. The first stage is to predict an arbitrary number of object proposals, and then in the second stage, they predict boxes by classifying and localizing those object proposals. However, these proposals have inherent problems of slow inference time, lack of global context (even within a single image), and complex architectures. Pioneer work with the two-stage approach is: Region-based fully convolution network (R-FCN) [6], Feature Pyramid Network (FPN) [23] and Mask R-CNN [12] which are built upon R-CNN [9] line of work. There's also a parallel work stream around Pseudo-LiDAR [36] in which dense depth is predicted in the first stage, thereby converting pixels into a pseudo-point-cloud. Then LiDAR-like detection head is applied for 3D object detection as done in Point-pillars [19].

3.2. Single-stage Detectors

YOLO [30] and SSD [24] opened the gate for single-stage detectors. These detectors classify and localize semantic objects in a single shot using dense predictions. However, they rely heavily on the post-processing Non-maximum Suppression (NMS) step to filter out duplicate predictions as one of the over-head. Their dependence on anchor box heuristics was addressed in Fully Convolutional One-Stage Object Detection (FCOS) [32] to predict 2D boxes based on center-pixel. The extension of this work is seen in FCOS3D [35], where they address the 3D object detection problem by regressing 3D parameters per object. These methods still heavily rely on post-processing for duplicate detections with NMS.

3.3. Set-based Detectors

This approach removes hand-designed NMS using set-based global loss that forces unique predictions per-object via bipartite matching. The pioneering paper, DETR [2], started this work chain. However, it suffers from slow convergence, limiting the features' spatial resolution. However, this issue was later addressed in Deformable-DETR [45] method, which replaces the original global dense attention with deformable attention that only attends to a small set of sampled features to lower the complexity and thereby speeds up the convergence. Another approach to accelerate convergence is SAM-DETR [43], which limits the search space for the attention module by using the most discriminative features for semantic-aligned matching. This line of work still has a CNN-based backbone. However, they use transformer [33] based detection head.

Above mentioned approaches operate per camera; however, the autonomous driving application needs to address the entire 360° scene, which includes $6 \sim 12$ surround cameras covering the entire spatial scene. Per-camera detections are generally aggregated using another set of NMS filtering to eliminate repeat detections originating from the camera overlap Field of View (FOV) regions. AVs need to maintain this long-range high FOV overlap to minimize the blind spots in the short range. Perspective view detections are lifted to BEV space by regressing depth per object or using the heuristic-based method, Inverse Perspective Mapping, by estimating ground-plane height.

4. Surround-Image Based Detectors

There are multiple applications of surround-camera-based computer-vision (CV) systems like surveillance, sports, education, mobile phones, and Autonomous Vehicles. Surround-view systems in sports are playing a huge role in the sports analytics industry. It lets us record the right moment across the field at the right moment with the right viewing angle. Surround-view vision has also spread its application in class monitoring systems, which lets teachers give personalized attention to each student. Nowadays, it is hard to find any smartphone with a single camera. However, to limit the scope of this paper, we will only focus on Autonomous driving-based computer vision.

A surround-view system uses features from different views to understand the holistic representation of the scene around the autonomous vehicle. Combining any two or more cameras requires prior infrastructure work related to fixed sensor mountings and their calibration. Calibration of the camera means extracting the extrinsic transformation matrix between the two cameras. This camera matrix enables us to make one-to-one mapping of a pixel in a camera to a pixel in another, creating a relation between multiple cameras to enable reasoning between themselves. Surround-view images can be represented by $\mathbf{I} \in \mathbb{R}^{\mathrm{N{\times}V{\times}H{\times}W{\times}3}}$. N, V, H, and W are the number of temporal frames, views, height, and width, respectively.

4.1. Why surround-view in an AV?

It is often hard to fit an entire object in a single frame to detect and classify it accurately. This is an especially common issue with the long-vehicle category. Let's take a visual understanding of what this means in as 5

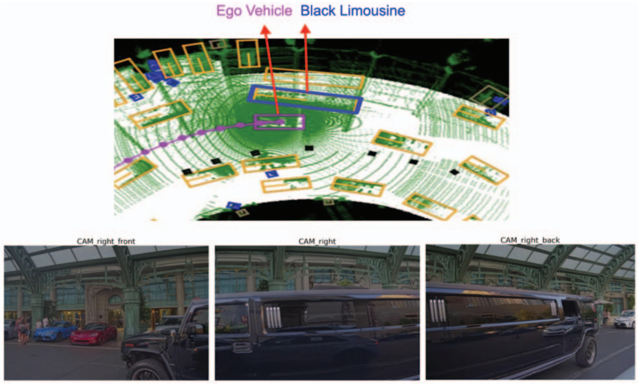

Figure 5. Usage of surround-view images in 3D object detection problem. BEV view (top); surround-view images of right-front, right, and right-back cameras (bottom). This shows that we may classify the object as a car with one or two cameras, but without all three images, we won't be able to perfectly localize, i.e., fit a bounding box on the black limousine.

4.2. Types of Surround-view Detectors

SoTA surround-view Detection can be broadly classified amongst two subgroups viz., *Geometry-based view transformers* and *Cross-attention-based vision-transformers*.

4.2.1 View Transformers

Pioneer work: Lift, Splat, Shoot [29] started a chain work where they *lift* each image individually into a frustum of BEV features, then *splat* all frustums onto a rasterized BEV grid. Given n images $\mathbf{X_k} \in \mathbb{R}^{3\mathrm{xHxW}}{}_{n}$, each with an extrinsic matrix $\mathbf{E_k} \in \mathbb{R}^{3\mathrm{x}4}$ and an intrinsic matrix $\mathbf{I_k} \in \mathbb{R}^{3\mathrm{x}3}$, we can find a rasterized BEV map of the feature in BEV coordinate frame as $\mathbf{y} \in \mathbb{R}^{\mathrm{CxXxY}}$, where C, X, and Y are channel depth, height and width of the BEV map. The extrinsic and intrinsic matrices define the mapping from reference coordinates (x, y, z) to local pixel coordinates (h, w, d) for each n camera. This approach does not require access to any depth sensor during training or testing; just 3D box annotations are enough. This architecture is demonstrated in as 6. One of the latest development in this line of work is BEVDet [16], which improves on pre-processing and post-processing techniques.

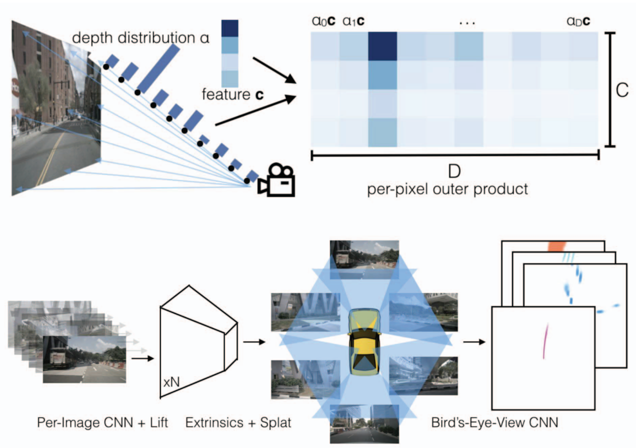

Figure 6. Lift-splat-shoot (LSS) [29] architecture: Lift step is visualized where per-image-frustum's pixel is projected to a discrete depth in BEV coordinate space with a context vector(top). The overall architecture is shown, which takes in n images and returns BEV semantic map (bottom).

BEVDet4D [14] adds temporal dimensionality to this method, making it a 4-dimensional problem. They tried to address the inherent problem of high-velocity error in vision-based detectors. Single-frame vision-based detectors generally have higher velocity errors than laser-based sensors, as LiDAR detectors use multiple-sweep data with temporal information embedded in the point cloud. RADAR's inherent point cloud includes velocity attributes using the Doppler effect. Adding temporal frames in a vision-detector enables us to learn temporal cues of the dynamic agent on the road.

As a further extension, BEVDepth [21] method adds a camera-aware depth estimation module that facilitates the object depth predicting capability. They hypothesize that *enhancing depth is the key to high-performance camera 3D detections* on nuScenes benchmark. They have replaced the vanilla segmentation head in LSS with the CenterPoint [42] head for 3D detection. They use supervision from the detection loss only for the auxiliary depth head baseline. However, due to the difficulty of monocular depth estimation, a sole detection loss is far from enough to supervise the depth module. Then used, calibrated LiDAR data was to project the point cloud onto the images using camera transformation matrices hence forming 2.5D image coordinates $P^{img}i(u, v, d)$, where u and v denote coordinates in pixel coordinates and d denotes depth from the corresponding LiDAR point cloud. To reduce memory usage, further development of M^2BEV [38] decreases the learnable parameters and achieves high efficiency on both inference speed and memory usage.

These detectors include four components: 1. An image encoder to extract the image features in a perspective view; 2. A depth module generates depth and context, then an outer product to get the point features; 3. A view transformer to convert the feature from perspective to BEV view; and lastly, 4. A 3D detection head to propose the final 3D bounding boxes. BEVStereo [20] introduces a dynamic temporal stereo method to enhance depth prediction within compute cost-budget. Simple-BEV [11] introduces RADAR point cloud on the LSS approach. Based on view transformers, BEVPoolv2 [15] is the current SOTA per nuScenes [1] vision-detection leaderboard. They use a BEVDet4D-based backbone with dense-depth and temporal information for training. They have shown *TensorRT* runtimes speedups as well. *TensorRT* is the model format generally used by *Nvidia* deployment hardware.

4.2.2 Vision Transformers

Vision Transformers can be divided as per the granularity of the queries (object proposals) in the transformer decoder as per [27] viz., sparse query-based and dense query-based methods. We will go into detail about the representative work for both of these categories:

Sparse Query-based ViT: In this line of work, we try to learn object proposals to look for in the scene from the representative training data and then use those learned object proposals to query at the test time. Here the assumption is made that test data objects are representative of the training data ones.

Seminal paper with single-image (Perspective-view), DETR [2] started this line of work, which is later extended to surround-view images in BEV space with DETR3D [37]. Here given n surround-view images $\mathbf{I} \in \mathbb{R}^{\text{H'×W'×3}}$, the backbone and/or FPN and/or Transformers encoder produce n encoded image features $\mathbf{F} \in \mathbb{R}^{\text{HWxd}}$, where d is the feature dimension, and H', W' and H, W denote spatial sizes of the image and the features, respectively. Then these n encoded features and a small set of object queries $\mathbf{Q} \in \mathbb{R}^{\text{Nxd}}$ are fed into the Transformer decoder to produce detection results. Here N is the number of the object queries, typically 300 $\sim$ 900 for the entire 360° scene as a meta-data camera; transformation matrices are also used as an input. These matrices are required to create 3D reference point mapping onto the 2D coordinate space and sample respective 2D features per query.

In the Transformers decoder, object queries are sequentially processed by a self-attention module, a cross-attention module, and a feed-forward network (FFN), and then finally by a Multi-Layer Perceptron (MLP) to produce 3D BEV detections as the final output. For an interpretation: object queries denote potential objects at different locations on the BEV map; the self-attention module performs message passing among different object queries; and in the cross-attention module, object queries first search for the corresponding regions/ views to match, then distill relevant features from the matched regions for the subsequent predictions. Also worth noting is that the transformer-based encoder is an optional add-on here, but the core part of these detectors is in the transformer-based decoders. The workflow of this approach can be easily understood as 7

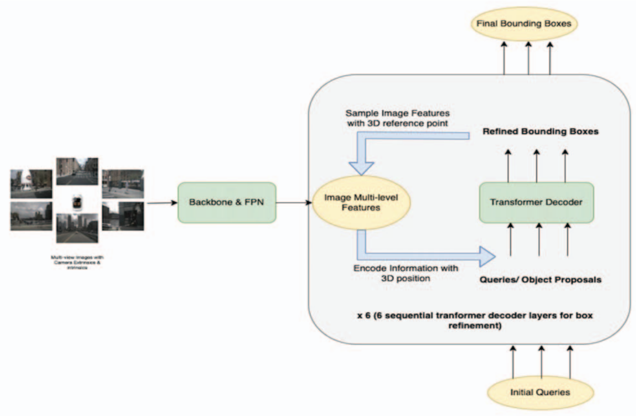
Final Bounding Boxes
Sample Image Features with 3D reference point
Refined Bounding Boxes
Backbone & FPN
Image Multi-level Features
Transformer Decoder
Multi-view images with Camera Extrinsics & intrinsics
Encode Information with 3D position
Queries/ Object Proposals
x 6 (6 sequential tranformer decoder layers for box refinement)
Initial Queries

Figure 7. Adaptation workflow from DETR3D [37].

As a further development of this work, Polar DETR [4] parameterizes 3D detections in polar coordinates, which reformulates position parametrization, velocity decomposition, perception range, label assignment, and loss function in the polar coordinate system (r, θ). This approach eases optimization and enables center-context feature aggregation to enhance the feature interaction. In Graph-DETR3D [5], they quantify the objects located at different regions and find that the "truncated instances" (i.e., at the border regions of each image) are the main bottleneck hindering the performance of DETR3D. Although it merges multiple features from two adjacent views in the overlapping regions, DETR3D[37] still suffers from insufficient feature aggregation, thus missing the chance to boost the detection performance fully. To address this issue, Graph-DETR3D[5] aggregates surround-view imagery information through graph structure learning (GSL). It constructs a dynamic 3D graph between each object query and 2D feature maps to enhance the object representations, especially at the image-border regions.

A positional encoding development work by PETR [25] cites a problem with the 2D encoding of features in the former approach. They transform surround-view features into a 3D domain by encoding the 3D coordinates from camera transformation matrices. Object queries can now be updated by interacting with the 3D position-aware features and generating 3D predictions, simplifying the procedure. A follow-up work PETRv2 [26] adds dimensionality to get temporal-aware denser features.

Dense Query-based ViT: Here we have a dense query based on the region of interest in the BEV representation. Each query is pre-allocated with a spatial location in 3D space. This line of work is better than the former in that we can still detect certain objects not learned as object proposals in the training data with sparse queries. In other words, *this approach is more robust to the scenario when training data is not the perfect representative of the test data*.

With this line of work, Pioneer was BEVFormer [22]. They exploit spatial and temporal information by interacting with spatial and temporal space through predefined grid-shaped BEV queries. To aggregate spatial information, they designed spatial cross-attention that each BEV query extracts from spatial features across the camera views. For temporal information, they use temporal self-attention to recurrently fuse the history BEV information as shown in 8. This approach at the time surpassed sparse-query-based Vision Transformers methods by getting higher recall values, owing to the fact of exploiting dense queries. However, dense queries come at the cost of high compute requirements, which was tried to address using deformable-DETR's [45] K-points around reference point sampling strategy. The fully transformer-based structure of BEVFormer makes its BEV features more versatile than other methods, easily supporting non-uniform and non-regular sampling grids.

A follow-up work, BEVFormerV2 [41] adds perspective supervision, which helps convergence and leverages image-based backbone better. This brings back two-stage detectors, where proposals from the perspective head are fed into the bird's-eye-view head for the final predictions. In addition to the perspective head proposals, they use DETR3D-style learned queries. For auxiliary perspective loss, they use FCOS3D [35] head which predicts the center location, size, orientation, and projected center-ness of the 3D bounding boxes. The auxiliary detection loss of this head, denoted as perspective loss L_{pers}, complements the BEV loss L_{bev}, facilitating the optimization of the backbone. The whole model is trained with a total objective

$$L_{total} = \lambda_{bev} L_{bev} + \lambda_{pers} L_{pers} \tag{4}$$

Table 2. Results of vision-only 3D object detections on nuScenes camera-only 3D detection benchmark on the test set. Abbreviations are defined in 5.

METHOD	YEAR	MAP	MATE	MASE	MAOE	MAVE	MAAE	NDS
BEVPOOLV2	2022	0.586	0.375	0.243	0.377	0.174	0.123	0.664
BEVFORMER V2	2022	0.580	0.448	0.262	0.342	0.238	0.128	0.648
BEVSTEREO	2022	0.525	0.431	0.246	0.358	0.357	0.138	0.610
BEVDEPTH	2022	0.503	0.445	0.245	0.378	0.320	0.126	0.600
POLARFORMER	2022	0.493	0.556	0.256	0.364	0.439	0.127	0.572
PETR V2	2022	0.490	0.561	0.243	0.361	0.343	0.120	0.582
BEVFORMER	2022	0.481	0.582	0.256	0.375	0.378	0.126	0.569
BEVDET4D	2022	0.451	0.511	0.241	0.386	0.301	0.121	0.569
GRAPH-DETR3D	2022	0.425	0.621	0.251	0.386	0.790	0.128	0.495
POLARDETR	2022	0.431	0.588	0.253	0.408	0.845	0.129	0.493
BEVDET	2021	0.424	0.524	0.242	0.373	0.950	0.148	0.488
PETR	2022	0.434	0.641	0.248	0.437	0.894	0.143	0.481
DETR3D	2021	0.412	0.641	0.255	0.394	0.845	0.133	0.479
FCOS3D	2021	0.358	0.690	0.249	0.452	1.434	0.124	0.428
CENTERNET	2019	0.338	0.658	0.255	0.629	1.629	0.142	0.400

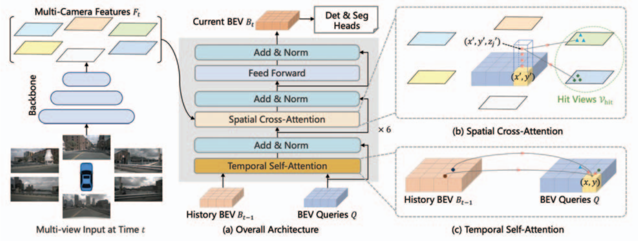

Figure 8. Overall architecture of BEVFormer [22]. (a) The encoder layer of BEVFormer contains grid-shaped BEV queries, temporal self-attention, and spatial cross-attention. (b) In spatial cross-attention, each BEV query only interacts with image features in the regions of interest. (c) In temporal self-attention, each BEV query interacts with two features: the BEV queries at the current timestamp and the BEV features at the previous timestamp.

PolarFormer [17] reasons the nature of the ego car's perspective, as each onboard camera perceives the world in the shape of a wedge intrinsic to the imaging geometry with the radical (non-perpendicular) axis. Hence they advocate exploiting the Polar coordinate system on top of the BEVFormer [22].

5. Experiments

nuScenes [1] is the widely used dataset in the literature for which sensor setup shown in 4 includes six calibrated cameras covering the entire 360° scene. Results on discussed pioneer works are shown on the test set of nuScenes in 2. This is under the filter *camera track detections*. The key for the metric abbreviations is as follows: mAP: mean Average Precision; mATE: mean Average Translation Error; mASE: mean Average Scale Error; mAOE: mean Average Orientation Error; mAVE: mean Average Velocity Error; mAAE: mean Average Attribute Error; NDS: nuScenes detection score.

6. Further Extensions

Based on the most recent developments around the surround-view BEV vision detections, we will now highlight possible future directions for the research.

Deployment compute-budget and run-time constraints: Autonomous vehicles operate on a tight compute budget, as we can have a limit of compute resources on board. However, when the 5G internet became mainstream, all computation could have been shifted to cloud computers. We, the industry as a whole, should start focusing on the run-time constraints of these compute-expensive transformer-based networks. One possible direction is to limit the object proposals (queries) based on input-scene constraints. However, there is a need to have a smart way to handle it, or else these networks may suffer through a low recall issue.

Smart object proposal initialization strategies: We may develop query initialization strategies that mix and match sparse and dense query initialization to enable both pros. The major con of the dense query-based approach is its high run time. This can be handled by using HD maps to focus only on the areas of the road which matter the most. Like BEVFormerv2 [41], object proposals can also be taken from different modalities. As one step further, these proposals may also be taken from the past time-step, with a fair assumption that the driving scene won't have changed much within a fraction of a second. However, to make AVs scalable, researchers need to focus more on affordable sensors like cameras and RADARs and not too much on expensive LiDARs or HD-Maps.

Collaborative Perception: A relatively new field of the area uses multi-agents and multi-view transformers to enable collaborative perception. This setup requires minimal infrastructure to enable smooth communications between different AVs on the road. CoBEVT [39] shows initial proof of how Vehicle-to-Vehicle communication may lead to superior perception performance. They test their performance on OPV2V [40] benchmark dataset for V2V perception.

7. Conclusion

We introduced development work around vision-based 3D object detection focused on autonomous vehicles in this work. We reviewed more than 60 papers and 5 benchmark datasets to prepare this paper.

Specifically, we first build a case on why a camera-based surround-view detection head is important for solving autonomous vehicles. Then we started with how research has progressed from single-view detection and extended to surround-view detection head paradigm, thereby increasing the detection performance. We have categorized two prominent categories for surround-view camera detectors to keep an eye on viz., Geometric View-Transformers based and *Vision Transformers based*. In the end, we proposed our take on surround-view detection trends, focusing on deploying those networks on an autonomous car, which may enlighten future research work.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *CoRR*, abs/2005.12872, 2020.

[3] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[4] Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Chang Huang, and Wenyu Liu. Polar parametrization for vision-based surround-view 3d detection, 2022.

[5] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinhong Jiang, and Feng Zhao. Graph-detr3d: Rethinking overlapping regions for multi-view 3d object detection, 2022.

[6] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. *CoRR*, abs/1605.06409, 2016.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[9] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[10] Ross B. Girshick, Forrest N. Iandola, Trevor Darrell, and Jitendra Malik. Deformable part models are convolutional neural networks. *CoRR*, abs/1409.5403, 2014.

[11] Adam W. Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-bev: What really matters for multi-sensor bev perception?, 2022.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[13] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset, 2020.

[14] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection, 2022.

[15] Junjie Huang and Guan Huang. Bevpoolv2: A cutting-edge implementation of bevdet toward deployment, 2022.

[16] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *CoRR*, abs/2112.11790, 2021.

[17] Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. Polarformer: Multi-camera 3d object detection with polar transformer, 2022.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[19] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds, 2018.

[20] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with dynamic temporal stereo, 2022.

[21] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection, 2022.

[22] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers, 2022.

[23] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016.

[24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[25] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection, 2022.

[26] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Aqi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petrv2: A unified framework for 3d perception from multi-camera images, 2022.

[27] Yuexin Ma, Tai Wang, Xuyang Bai, Huitong Yang, Yuenan Hou, Yaming Wang, Yu Qiao, Ruigang Yang, Dinesh Manocha, and Xinge Zhu. Vision-centric bev perception: A survey, 2022.

[28] Abhishek Patil, Srikanth Malla, Haiming Gang, and Yi-Ting Chen. The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. *2019 International Conference on Robotics and Automation (ICRA)*, pages 9552–9557, 2019.

[29] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. *CoRR*, abs/2008.05711, 2020.

[30] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[31] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *arxiv*, pages 2443–2451, 06 2020.

[32] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: fully convolutional one-stage object detection. *CoRR*, abs/1904.01355, 2019.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[34] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001.

[35] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: fully convolutional one-stage monocular 3d object detection. *CoRR*, abs/2104.10956, 2021.

[36] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving, 2018.

[37] Yue Wang, Vitor Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. DETR3D: 3d object detection from multi-view images via 3d-to-2d queries. *CoRR*, abs/2110.06922, 2021.

[38] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M. Alvarez. M^2bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation, 2022.

[39] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers, 2022.

[40] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication, 2021.

[41] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, Jie Zhou, and Jifeng Dai. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision, 2022.

[42] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CoRR*, abs/2006.11275, 2020.

[43] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. Accelerating detr convergence via semantic-aligned matching, 2022.

[44] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International journal of computer vision*, 73(2):213–238, 2007.

[45] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. *CoRR*, abs/2010.04159, 2020.