# BEV感知相关问题

## 1. 技术优越性

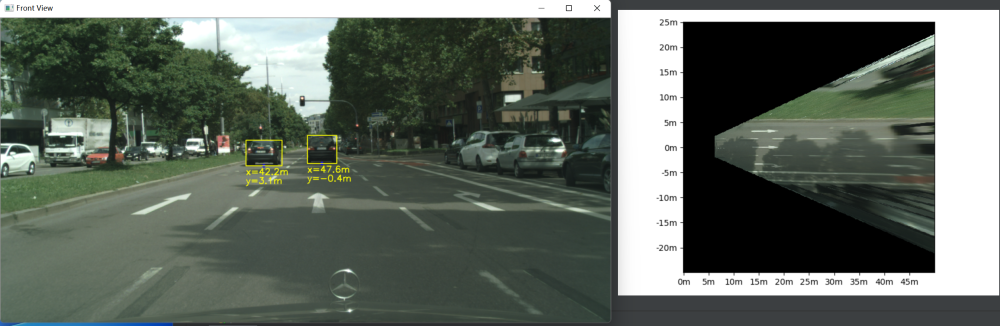
### - 多视角比单目/双目的方案好在哪（效果上）

**单目:** 这里的单目指的是单视角输入，一般用于2D的相关视觉任务，如2D目标检测、车道线检测等任务

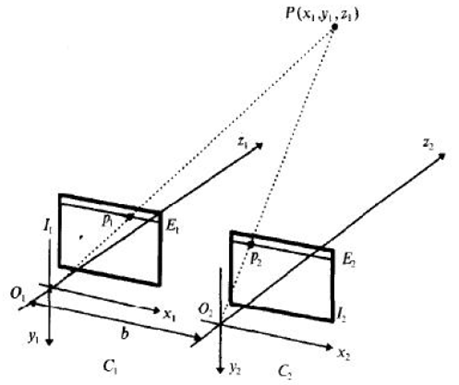
* 在自驾领域基本都是3D下的视觉感知，在2D图像上视觉任务的效果再好，变换到3D视角下也会存在问题，如车道线检测，由于远处视角下的像素点较少，变换到BEV视角会出现噪声比较多的问题，如图所示，变换之后噪声较多，且经过优化之后，车道线部分弯曲



* 单目对于3D感知中的一个重要任务-测距，无法比较好的实现，传统的在2D视角进行距离估计的方法如，基于IPM将2D图片变换为BEV视角，需要在地面是一个平面的基础上才能预测准确，见下图所示，再如对2D图像进行深度估计，直接输出图像中的每个像素点的距离，类似于语义分割任务，数据难获取，且极易受到环境的影响，长尾问题严重



**双目:** 使用双目的方式可以实现测距



* 双目测距的方法在于从两张图像中找到匹配的像素点，再计算距离，这里容易出现误匹配且计算量巨大

* 双目相机测距的精度与基线(图中的b)的长度有关，基线越长，测距的精度越高，但目前的基线不够长，不太适用于自动驾驶领域

单目与双目的一个共有问题在于只能感知车辆前方的信息，对于车辆两边和后方无法进行感知，对于后续的规划控制来说存在较大的风险，如某个场景下需要进行变道超车，这里就需要感知当前所属车道和左右两侧是否有车辆或后方是否有车正在进行超车，使用**多视角输入**，可以全面感知车辆周边信息，针对不同路况做出正确的规划控制

#### 多目是否要求摄像头的规格均一致？

多目不要求摄像头的规格一致，只要保证各摄像头之间的视场角交叉覆盖即可，使用不同参数的摄像头，在将特征离散到BEV视角时，使用不同的相机内参进行变换，理论上并不会有影响，但对于模型来说会加大学习难度，模型需要根据不同的摄像头类型，再进行深度分布学习时，不同内参的摄像头学习的深度分布不一样，极大的增加了学习难度，如前视相机与左侧视角相机在车辆行驶过程中可能会拍摄到同一区域，由于相机内参存在较大差别，学习到的特征分布与深度分布均有较大差别，这里就需要更多的数据才能让模型学习好

#### 届于单目和多目之间的伪多目方案，比如知行科技的4R5V方案（1前视+4环视）对比如何？

使用一个前视+4环视的方案，是从APA的技术方案中改变而来，APA场景中，对于相机的要求是能准确识别，车身旁边的停车位，范围一般在5-10米内，因此对于环视相机来说，只能看到车身周边较近的范围，只适用于低速无人驾驶场景(如APA场景、园区接驳车等)或者简单的高速驾驶场景，但对于大部分复杂的自动驾驶场景，需要车辆对车身周边更远的信息作出感知，以便更好进行规划控制，

### - 同样多视角输入，基于BEV方案比各个单路做然后融合的方案效果好在哪

#### summary：

基于BEV的视角方案，首先会对车身环绕摄像头采集到的摄像头进行特征提取（与传统的2D CNN一样），然后对提取到的特征进行深度分布预测并变换至BEV视角，再在BEV视角下进行视觉感知任务预测，因此基于BEV视角方案既可以用于2D目标检测/分类/分割任务如traffic light及其属性识别、2D车辆及其属性识别、行人及其属性识别、2D道路事件感知等，还可以用于3D视角下的空间位置定位学习，如3D目标检测(行人、车辆等)、BEV视角下的可行驶区域识别、车道线识别等多种3D任务

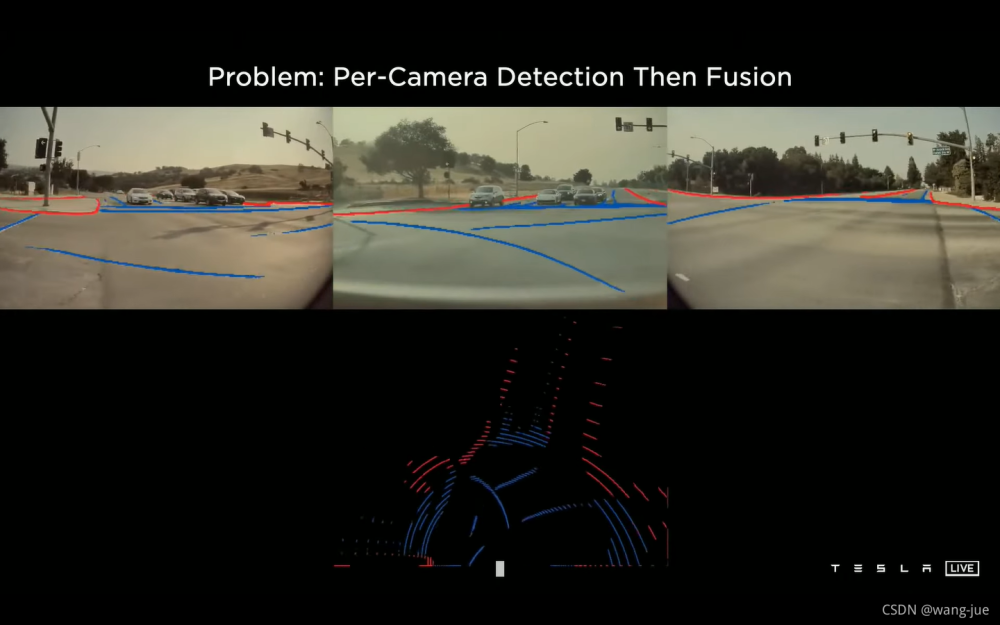
* **自动驾驶本身就是一个3D/BEV预测的问题，所有的2D结果在最终规划控制时还是需要转换到BEV视角下**

* **多视角后融合对于大目标(各视角只看到部分目标)的检测后融合处理复杂且很难区分是否同一目标**

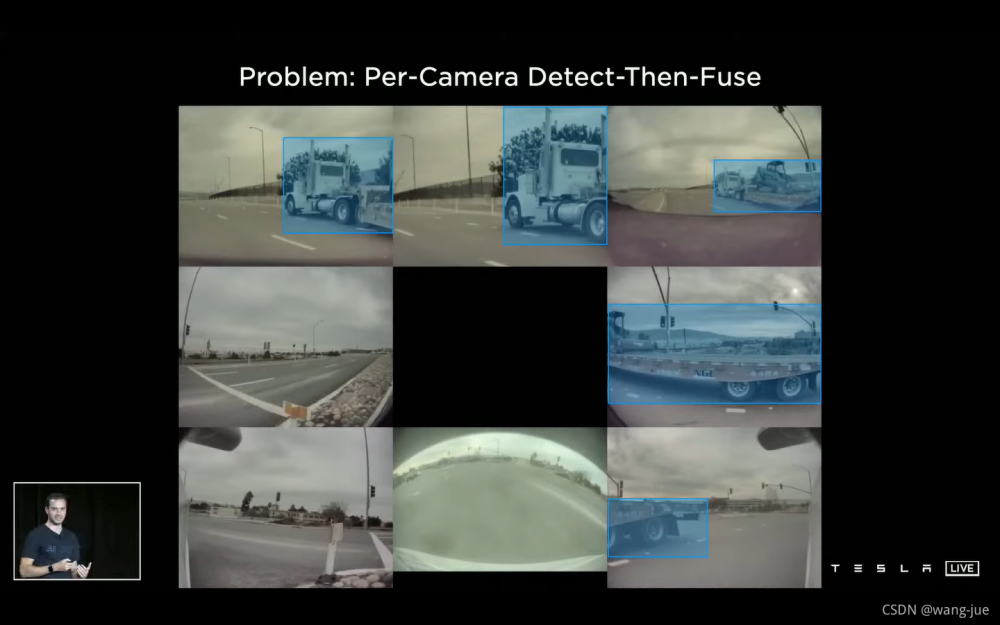
* **多视角后融合对于车道线检测极易受到误差干扰，模型学习时无法学习各视角下的车道线的共线约束关系**

**多视角后融合方案-参照Tesla AI Day**

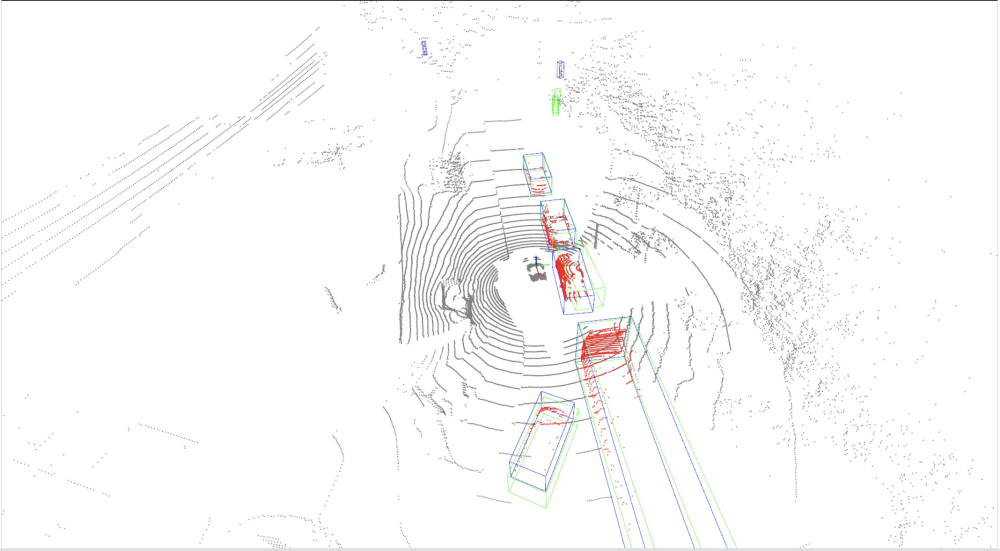
车道线检测: 对于各个相机视角下的车道线检测结果，融合后存在较大误差，离相机距离近的特征点密集，远的离散，导致预测车道线不连续



目标检测:对于2D视角下的目标检测，不同图像拍摄到的都是目标的局部信息，无法判断属于同一目标



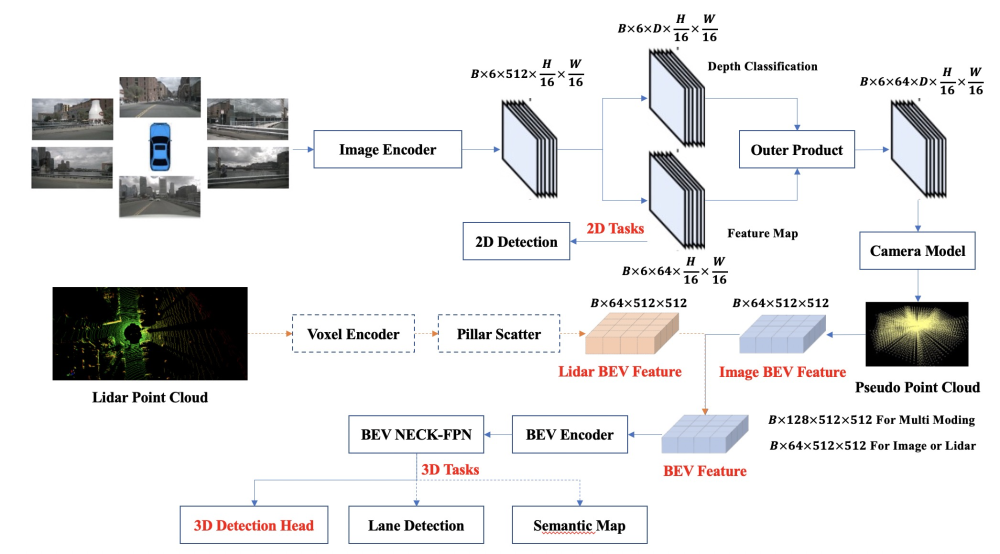
BEV视角前融合: 基于BEV视角下的3D目标检测，先将多目的视觉特征进行融合并变换至BEV视角下，再在一个完整BEV视角下进行目标检测，避免了同一目标无法判断的问题，如下图所示，在BEV视角下检测的结果再变换到不同相机视角下的效果





### - 从下游任务拓展或Lidar融合来说，基于BEV的视觉方案好在哪

现有的BEV视角方案:



* 现有的BEV视角下的感知模型，如上图所示，可以在纯视觉感知模块后面接入2D视觉感知任务，如红绿灯、标识牌等识别任务，在BEV NECK-FPN后面接入3D视觉任务，如3D目标检测、车道线检测等任务，整体架构可以接入不同的任务模块，最终统一至一个神经网络架构中。

* 同时对于Lidar感知层面，Lidar采集的点云通过PointPillars等模型可以很方便的转换为BEV架构，本身Lidar采集的数据就很容易转换到BEV视角，对点云进行特征编码之后，可以很好的与Image BEV进行特征融合，可以直接进行concat，也可以使用Transformer的self-attention机制来进行深层次的特征融合

* 现有的框架可以进行独立模块的升级优化，如Image BEV视角除了现有的基于相机参数进行跨域转换之外，也可以替换为基于Tranformer cross-attention机制来进行跨域生成，这也是Transformer本身的优势所在，使用Transformer来让模型自己去学习到视角特征的转换，相比于基于相机外参的方法具有一定的优势，鲁棒性也会更强一些

* 同时在BEV视角下学习到的特征包含了运动目标的速度、方向等信息，可以很方便的融合时序信息，将不同时刻的BEV视角进行对齐后融合，可以进一步增强目标速度预测的精确度

#### BEV 不等于 Transformer，用不用transformer的差异？？

BEV使用Transformer和不使用Transformer的主要区别在于几点:

* 在生成Image BEV Feature时，直接基于相机内外参来将基于深度离散化的特征变换到BEV视角下特征，而基于Transformer的方式是使用Object Query来代表BEV视角下各个点要学习的特征(类似Anchor-MDETR)，让模型自己学习如何去融合关联特征

* 如果使用Lidar点云的特征时，不使用Transformer的BEV视角进行图像特征和点云特征融合时，一般是直接concat，再送到MLP进行深层次融合，而基于Transformer的BEV对不同模态特征进行融合，一般基于self-attention机制进行融合(deformable conv)

* 对于需要融合时序特征的情况(如进行动态目标的速度预测，便于目标跟踪)，需要将不同时刻的BEV特征进行融合，不使用Transformer的BEV，首先需要对BEV视角进行特征对齐后再进行融合，使用Transformer的BEV则可以同样基于self-attention来进行自我学习

总结来说:

BEV Without Transformer: 直接基于相机的内外参来将视觉语义特征变换到BEV视角特征，依赖相机内外参标定的准确性，算力相比于Transformer要低不少，需要的数据量少于Transformer BEV

BEV With Transformer: 更优的特征融合方法(时序、多模态等)，但需要更多数据来训练Transformer架构，同时算力要求也更高些

### - 缺点在哪（比如性能等）

现有方案的缺点

* 成本肯定是比单目的成本高，因此所需要的算力也会随着摄像头的增多而增加 --> 具体对比：单目 vs 传统多目 vs BEV 多目？

* + 假设不同方案原始图像的输入分辨率均为1600\*900，统计结果来源于mmdetection框架的get\_flops，拉到同一个尺寸进行比较，如下表所示。总结来说，传统多目与BEV多目算力上基本上持平，模型架构消耗算力最多的部分在于对多目输入的图像进行特征提取时所消耗的算力，这与DETR3D的算力评估基本一样，<https://yuque.antfin.com/zppdqz/ad0heo/em9y6q>

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Models** | **Input Size** | **Image Backbone** | **Image Neck** | **Image Head** | **Params**  **(MB)** | **Image Encoder**  **GFlops** | **BEV Size** | **BEV**  **Backbone** | **BEV**  **Neck** | **Params**  **(MB)** | **BEV**  **GFlops** | **Image Encoder** | |
| **GFlops** | **比重** |
| 单目 | (1, 3, 900, 1600) | ResNet50 | DilatedEncoder | YOLOFHead | 43.88 | 142.82 | / | / | / | / | / | 142.82 | 100% |
| 传统多目 | (6, 3, 900, 1600) | ResNet50 | DilatedEncoder | YOLOFHead | 43.88 | 856.92 | / | / | / | / | / | 856.92 | 100% |
| BEV多目 | (6, 3, 900, 1600) | ResNet50 | DilatedEncoder | / | 41.96 | 840.12 | (128, 512, 512) | ResNet50 | FPN | 27.22 | 49.34 | 889.46 | 94.4% |

* 多目摄像头之间的标定方法，多目相机的外参的准确性严重影响了BEV视角特征的生成质量 --> 跟上面优点矛盾？

* + 不同的标定的算法会导致多目摄像头之间的内外参存在一定的差别，如何进行多目相机与激光雷达之前的时间同步需要硬件层面的支持，在不使用Transformer的BEV方案下，模型依赖相机内外参来进行BEV视角特征生成，虽然模型的学习能力能一定程度的弥补参数的误差，但还是需要更多的数据来使模型学习到这方面的能力

* 基于Transformer的方案可能需要较大量的数据来进行训练才能得到比较好的效果 --> 数据和标注

* + 现在的自驾驾驶标注方式，基本都是借助Lidar的信息来进行数据标注，特别是3D场景下的数据标注，此时多目相机借助Lidar来进行标注就会导致成本比较高。对于单目来说，不光有很多开源数据可用，且标注的成本也低。

## 2. 基于上述优缺点的梳理，和我们过去一年在VQA的沉淀，我们来做BEV方案的优势在哪

**目前团队已有沉淀与BEV方案的优势:**

* **对不同模态数据的特征融合方法**: 已经在VQA任务进行过较多的探索，如使用Transformer/CLIP来对不同模态的数据进行高维空间语义特征对齐，将不同模态的数据映射至高维空间中距离相近的位置，再对融合后的特征用于下游任务。Lidar点云与图像特征也属于不同模态的数据，如何对其进行有效的融合是影响自驾效果的关键一环

* **不同域数据的生成方法**: 对于BEV视角下的感知算法，涉及到了图像视角特征至BEV视角特征的跨域生成，与VQA任务中的将图像文本融合特征变换到Anchor特征的跨域任务使用的模型架构基本一致，对于如何基于Object Query进行优化改进并在多模态任务下实现目标检测是去年重要的技术沉淀

* **时序信息的特征融合**: BEV视角下的特征融合，由于需要先进对齐，再进行融合会有降低不匹配特征进行融合的概率，可以使用可变形卷积的方法来让模型自己学习如何根据相关特征的相关性进行特征融合，在VQA任务中，我们优化的Deformerable-MDETR架构也是基于这种思想来对模型进行优化的

* **端侧部署**: 在VQA任务中，为了能够将业务落地，做了大量的模型优化和轻量化部署的工作，最终将Transformer架构优化到可以在8155平台上达到准实时级的应用效果，BEV方案也会涉及到大量图像需要处理，多个op不支持的情况，在这方面也有较多的技术积累

具体一点？

端侧优化部署:<https://yuque.antfin.com/zppdqz/znp3ig/mag2m6>

[BEV与VQA方案对比.pptx](https://yuque.antfin.com/attachments/lark/0/2022/pptx/21256453/1654509093962-cd655627-2c61-4b70-bef8-a849c86beb3e.pptx)