Occupancy综述

引言

随着自动驾驶的不断发展，对周围环境的感知的重要性不断增强。自动驾驶场景往往输入信息只有多机位摄像头（往往是六个）和传感器的位姿信息，这相比于存在Lidar信息的场景感知难度有所提升。以往的方法往往使用BEV[1]或者3D Box，这些方法相比于Occupancy Grid Mapping都存在着一些缺陷，比如BEV丢失了高度方面的信息，3DBox对不规则物体的处理欠佳。而Occupancy的引入使得感知的精度增加，但也存在着训练久，运行慢的缺点待解决。

相关工作

数据集：在Occupancy任务中，网上现有的数据集具备的信息往往为摄像头，Lidar点云和位姿。将此类数据集转化成Occupancy数据集需要将稀疏的点云图密集化以及语义标签的自动生成。Scene as OCC[3]提出了一种benchmark:openOCC，基于数据集nuScenes,将Lidar的点云拆分成前景和背景，在世界坐标系将静态背景进行累加，在对象坐标系中对运动前景进行累加，以达到更为密集的效果。Occ3D[2]同样也提出了一个benchmark，采用半标注技术生成语义信息，需要人工标注的框形语义信息和可选的点级语义信息，相比于Scene as OCC,额外提出了遮挡推理，通过射线法判断体素是否可见，并且设置了”未知“掩码，单独设置未知标签。并且提出了基于图像的体素细化，连接图像原点和占用体素，找到与图像语义标签相同的第一个体素，并设置射线穿过的之前所有体素为空。增强的体素的细化度。

模型：VoxFormer[4]使用单摄像机视图首先将图像信息输入深度识别网络后得到深度信息，再前向映射成三维点云后和一个分辨率较低的Query进行点乘得到一个稀疏张量，最后和Image Feature做DFA。

Scene as OCC[3]提出了一种OccNet网络，使用BEV特征并且串联式提升为3D特征，同样使用Deformable attention。

Occ 3D[2]提出了一种从粗略到精细占用网络CTF-Occ，使用了Multi-Scaled的图像特征，从小到大逐步扩大3D体素分辨率，并且使用Token Selection，用额外的标签判别器判断某点体素是否为空，只留下来最不确定的K个体素。具有摆脱BEV压缩高度限制、对小物体识别精度更高的优势。

OccFormer[5]将深度信息与上下文信息分开形成Dual-path，分为Local和Global两部分，使用共享的Windowed Attention和ASPP网络，并且在bakcbone后面额外加了一个neck网络，得到的Depth Distribution和Context Feature做乘法，经过体素池化后输入Decoder中。Decoder使用了语义3D Mask思路。除此之外，该论文还提出了几点优化方案：

- Preserve Pooling，对于mask的降维处理时，不使用以往模型的双线性插值，而是使用最大池化，有效保留了局部特征。
- Class-Guided Sampling，在计算损失的时候，不使用均匀采样，而是估计训练集中的样本频率，样本频率越少，对损失的采样频率越高。以达到对各类的预测大致平衡的水平，防止出现偏向于预测频率较高的类的情况。

  Surround Occ[6]使用多个摄像机的多尺度特征，并且设计了2D to 3D架构，模仿BEV的backward方法推广到3D，用Spatial Cross Attention，对于每一个体素只关心自己投影到

的2D图像，解决了视图平均关注所导致的效果差问题。最后的多尺度信息中，小尺度信息会反卷积加到大尺度信息上，并且每个尺度都要预测，同时计算损失。

FB-Occ[7]结合了Forward(图片投影点云)和Backward(体素投影像素)两个方法，利用Depth Distance进行点云前向投影，并且提出了Depth-Aware的Backward Projection,将前向投影的点云拍扁后对每一个像素反向投影，结合深度信息产生BEV特征，之后沿高度展开，补全稀疏点云得到3D体素。

本文提出了2D转3D中backbone容易过拟合的问题，并且提出了相应的解决方案：深度和语义结合的预训练。将深度信息单独进行有监督学习，为了防止模型过于偏向深度信息导致丢失语义先验，作者使用SAM模型进行了自动标记，用框提示生成高质量的语义掩码。

SelfOcc[8]创新地摆脱了Lidar标注的需求，只依靠视觉图像进行自监督学习，相比于之前的3D无监督学习都使用2D图像进行NeRF,SelfOcc将图像特征直接放在了3D,强调了3D重建的重要性。同时改进了之前研究使用深度信息为辅助，本文使用了Multi-View Stereo模块包含的丰富深度信息指导NeRF，使其收敛速度大大提升。

OctreeOcc[9]对场景进行了统计研究，发现95.1%的部分都是Empty，说明了有大量体素都是被浪费的。本文引入了八叉树3D概念，使用八叉树对时间以及空间复杂度进行优化，达到了多颗粒度的效果以及24%的开销减少。利用多层的Octree查询不断分裂下去，最后经过解码后加上语义信息。最终损失里面应当包含八叉树结构Loss.

技术细节

- 关于Attention模块，大部分Occ网络使用的都是Deformable Attention,公式如下：

$$\text{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}) \big], \quad (2)$$

3D版本

$$\text{3D-DA}(\boldsymbol{q}, \boldsymbol{p}, V'_{t,i}) = \sum_{m=1}^{M} W_m \sum_{k=1}^{K} A_{mk} W'_k V'_{t,i}(\boldsymbol{p} + \Delta \boldsymbol{p}_{mk}),$$

也类似：这两个公式中K都远小于总体素的数量。

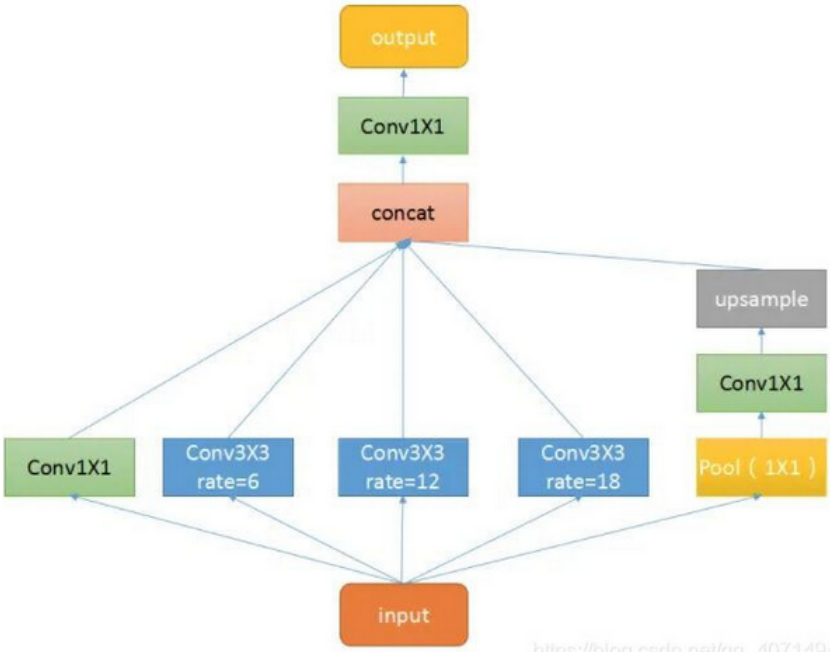
部分还会用到交叉可变形注意力DCA，公式如下：

$$\texttt{DCA}(\mathbf{q}_p, \mathbf{F}^{2D}) = \frac{1}{|\mathcal{V}_t|} \sum_{t \in \mathcal{V}_t} \texttt{DA}(\mathbf{q}_p, \mathcal{P}(\mathbf{p}, t), \mathbf{F}_t^{2D}), \quad (4)$$

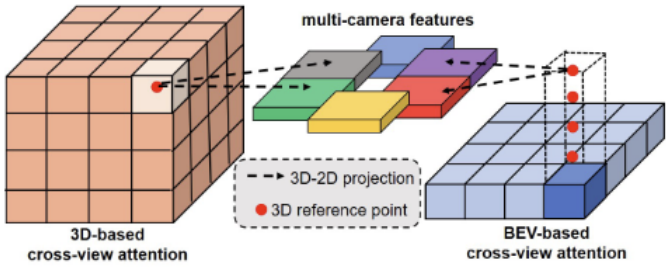
where t indexes the images, and for each query proposal $\mathbf{q}_p$ located at $\mathbf{p} = (x, y, z)$, we use camera projection function $\mathcal{P}(\mathbf{p}, t)$ to obtain the reference point on image t.

其中P表示3D向2D的投影。

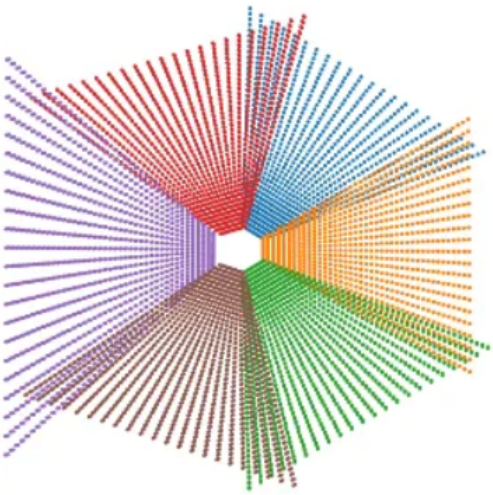
- OccFormer中用到的ASPP技术架构如下：

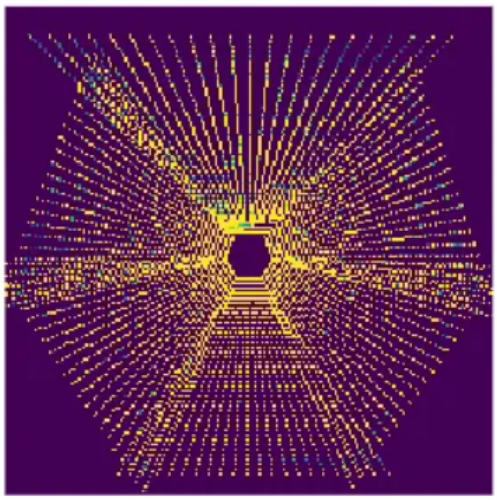


- Backward Projection可以类似如图的方式实现（左侧是体素backward，右侧是BEVbackward）：



- 前向投影原理如图：





- 大部分模型用到的transformer架构如图：

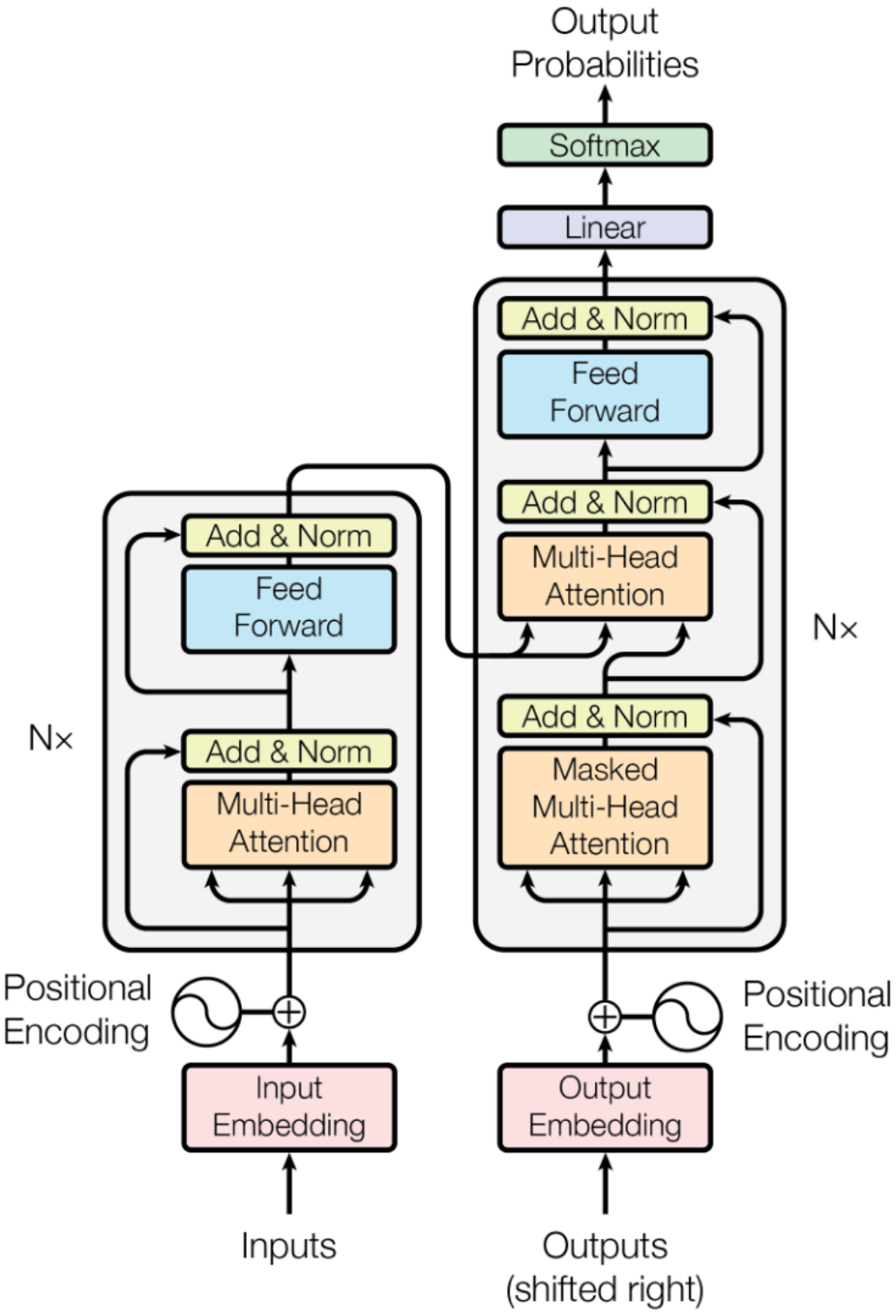


Figure 1: The Transformer - model architecture.

注：大部分都只用了Encoder.

参考文献

[1]BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers.

[2]Occ3D: A Large-Scale 3D Occupancy Prediction Benchmark for Autonomous Driving
[3]Scene as Occupancy
[4]VoxFormer: Sparse Voxel Transformer for Camera-based 3D Semantic Scene Completion
[5]OccFormer: Dual-path Transformer for Vision-based 3D Semantic Occupancy Prediction
[6]SurroundOcc: Multi-Camera 3D Occupancy Prediction for Autonomous Driving
[7]FB-OCC: 3D Occupancy Prediction based on Forward-Backward View Transformation
[8]SelfOcc: Self-Supervised Vision-Based 3D Occupancy Prediction
[9]OctreeOcc: Efficient and Multi-Granularity Occupancy Prediction Using Octree Queries