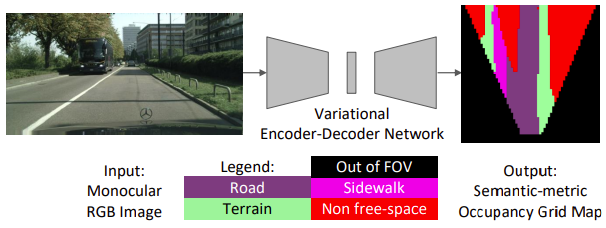
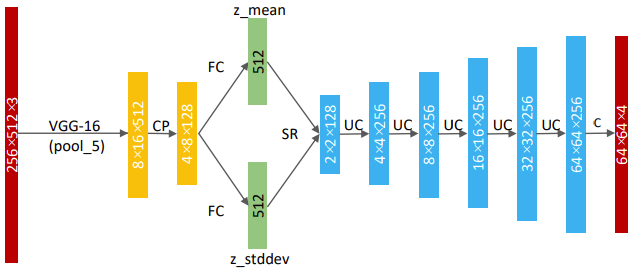
Monocular BEV Segmentation Review

**1.VED (** [**https://arxiv.org/pdf/1804.02176.pdf**](https://arxiv.org/pdf/1804.02176.pdf) **)**

VED网络的输入是一幅前视RGB图像，输出是一幅顶视占据栅格地图，其中的每个格子都会被分配一个语义类别，总共有四个类别（road, sidewalk, terrain, and non free-space）。

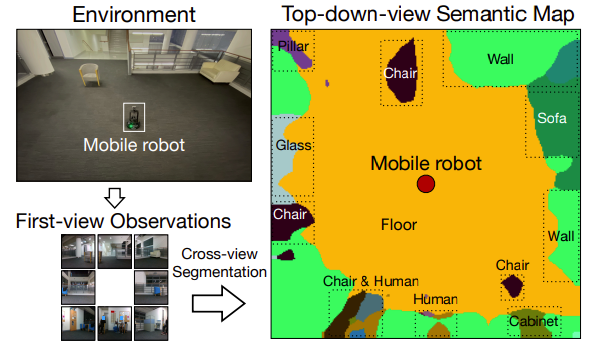


VED网络是一个典型的encoder-decoder结构，首先将前视图像编码成高维向量，然后解码这个高维向量得到顶视语义分割结果（很神奇有木有），最后通过网格化结合投票策略生成语义占据栅格地图。其中，用到了variational autoencoder的核心思想，即让隐变量符合标准正态分布。因此，并没有把encoder输出的向量直接作为隐变量，而是在encoder之后先通过全连接层生成均值向量和标准差向量，再通过这两个统计量来合成得到隐变量。作者通过实验证实variational sampling（variational autoencoder中的参数重构技巧）对扰动具有更好的鲁棒性，关于VAE的介绍可以参考<https://zhuanlan.zhihu.com/p/27549418>。

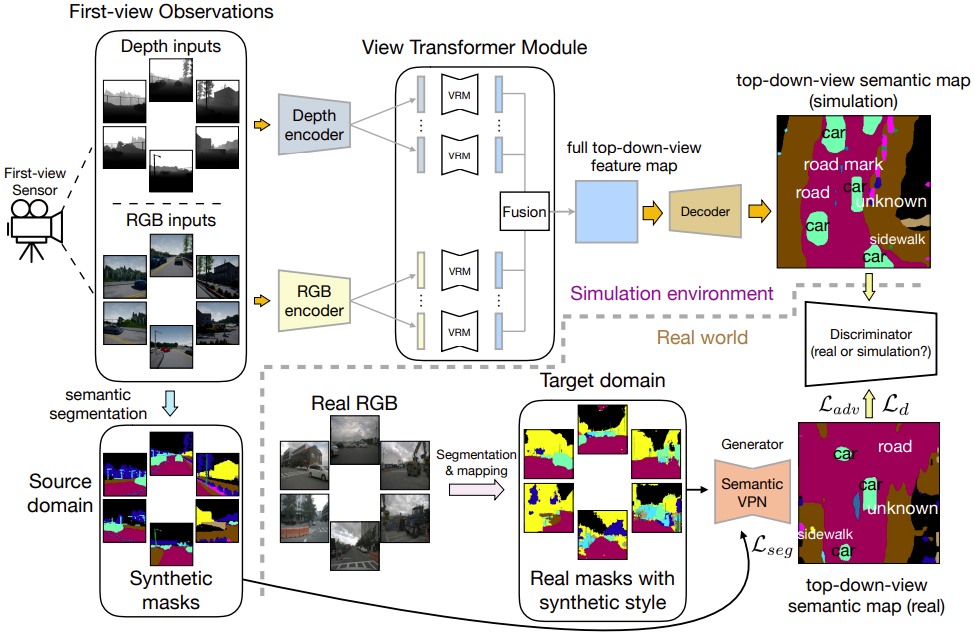


**2.VPN (** [**https://arxiv.org/pdf/1906.03560.pdf**](https://arxiv.org/pdf/1906.03560.pdf) **)**

VPN网络的输入是多视角多模态的传感器数据（RGB和Depth），输出是顶视的语义分割结果。作者是在合成数据集上训练得到的模型，为了提高模型在真实数据上的泛化性，作者采用了两种domain adaptation策略。第一种是像素级的adaptation，第二种是输出空间的adaptation，前者是为了让真实数据更加具有合成的风格，考虑到语义mask没有纹理差异同时也包含足够的信息，因此像素级的adaptation是在语义mask层面上进行的，后者采用对抗训练的方式使得在真实数据上生成的顶视语义分割结果更加精确。generator的loss包括分割loss和对抗loss，discriminator的loss用来做source与target的二元分类。



个人理解，文章的创新点可能就在于使用view transformer module将feature map从透视空间变换到bev空间，具体做法如下：对于每个视角每个模态，首先将经过encoder得到的feature map拉平成HWxC，然后通过一个MLP构建bev空间中某个pixel与透视空间中所有pixel之间的关系（bev空间的feature map与透视空间的feature map具有同样的形状，所以需要HW个MLP），接着将得到的feature map变形成HxWxC即为top-view feature map，最后将所有视角所有模态的top-view feature map相加即可得到full top-view feature map。这里有个疑问就是，为什么bev feature map需要与原始perspective feature map具有同样的形状？

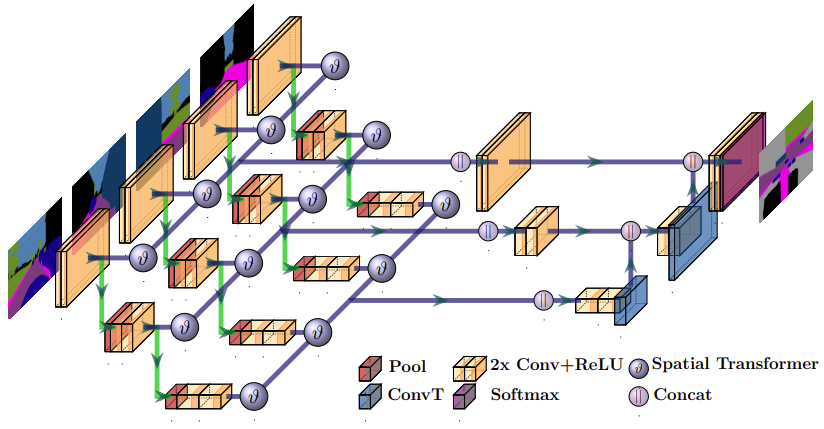


**3.Cam2BEV (** [**https://arxiv.org/pdf/2005.04078.pdf**](https://arxiv.org/pdf/2005.04078.pdf) **)**

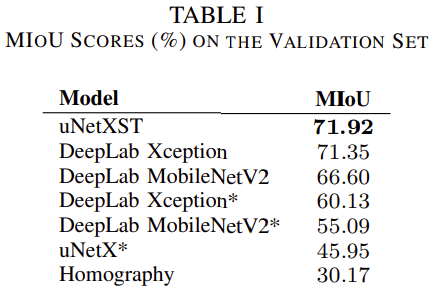
忍不住吐槽一下，这篇文章的写作逻辑真是混乱。作为读书笔记，先来梳理下整个pipeline，再对一些细节进行分析。文章提出了两个model，并对比了两个model的性能。

第一个是single-input model，输入是经过IPM变换后的拼接mask即文章中提到的homography image，输出是bev mask。模型用的是DeepLabv3+，本质上是利用神经网络对经过IPM变换后的拼接mask进行校正。那么这个homography image是怎么得到的呢？阅读[这部分代码](https://github.com/ika-rwth-aachen/Cam2BEV/tree/3071ed38251699b513517a5d3bee64f07bf9a13a/preprocessing/ipm)可以知道，首先通过IPM对每个相机的语义分割图像单独进行变换，然后将这些变换结果合并成单个图像即可。在合并的过程中，对于重叠区域（两个相机均可见的区域）中的像素，是从两个变换结果中随机选择的。

第二个是multi-input model，输入是多个视角的语义分割图像，输出是bev mask。作者在UNet基础上进行了扩展，使其支持多个输入图像，作者将其命名为uNetXST，本质上是对特征进行IPM。首先，对于每个视角的语义分割图像，都单独采用一个encoder提取多尺度特征。然后，将同一尺度下每个视角的特征通过各自的IPM进行特征的变换并concatenate起来。接着，低分辨率的特征经过上采样后通过skip-connection与高分辨率的特征进行融合。最后，对融合得到的特征进行softmax得到bev mask。个人觉得遗憾的是，在特征变换这一块并没有特别出彩的地方，只是简单地使用了一个预先知道参数的IPM变换。



实验结果也非常有趣，表中带星号的model表示在对应的model中移除IPM模块，对于single-input model意味着将多个相机的语义分割图像在channel维度上concatenate起来，对于multi-input model意味着禁用Spatial Transformer单元，Homography即为前面的homography image。



仔细分析表1中的数据，可以得到以下结论：

(1)对特征进行IPM变换相对于对语义分割mask进行IPM变换会有更好的效果，IPM会引入透视误差，而特征相对于语义分割mask对这种误差具有更好的包容性。

(2)IPM变换的确可以改善性能，这个是显然的。

一些细节：

(1)数据集是在仿真环境Virtual Test Drive中生成的，ego-vehicle配备有四个虚拟的广角相机覆盖360度环视以及一个虚拟的无人机相机记录bev ground-truth。

(2)对于被遮挡区域，作者设计了一套规则对bev ground-truth进行处理，并附加一个语义类别occlusion，具体细节可以参考[这部分代码](https://github.com/ika-rwth-aachen/Cam2BEV/tree/3071ed38251699b513517a5d3bee64f07bf9a13a/preprocessing/occlusion)。

(3)本文所有的输入图像都是经过语义分割后得到的mask，这样处理是为了让模型从仿真数据到真实数据的迁移能力更好，这一点在VPN与BEV-Seg中都有提及，直观上来看是由于语义分割后的mask剔除了一些domain-dependent特征如天气状况、采集时间（白天还是夜晚）等。

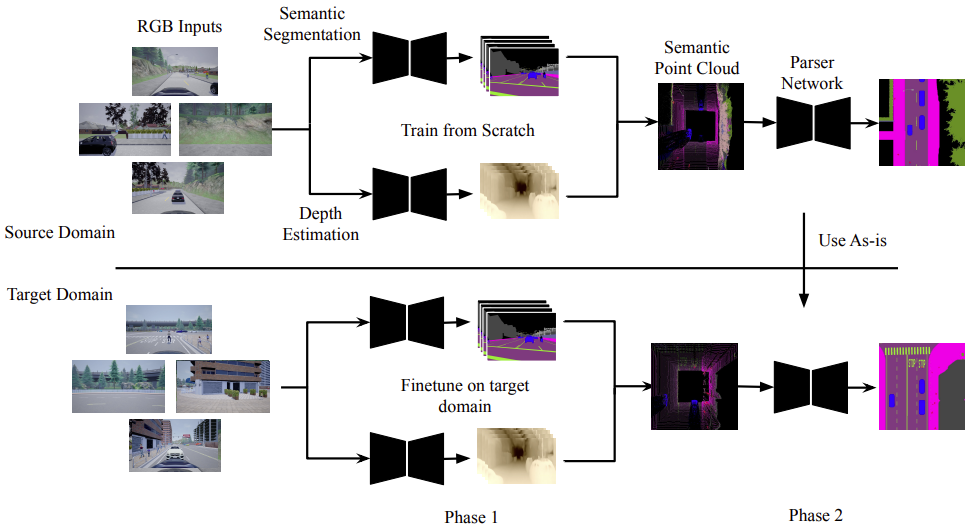
**4.BEV-Seg (** [**https://arxiv.org/pdf/2006.11436.pdf**](https://arxiv.org/pdf/2006.11436.pdf)) **)**

这篇文章没有涉及到feature从perspective view到bev的变换，而是直接生成pseudo-lidar point cloud。具体而言，整个pipeline包括两个阶段。在第一个阶段，对于每个视角的输入图像，首先通过语义分割分支和深度估计分支得到各自的segmentation map和depth map，然后基于针孔相机模型得到各自相机坐标系下的semantic point cloud，接着借助相机外参将这些semantic point cloud转换到车辆坐标系下并concatenate起来得到一个大的semantic point cloud，最后就是将这个大的semantic point cloud投影到bev，然而这样得到bev semantic segmentation会存在一个问题，就是很多像素点是没有分配到语义label的。于是，在第二个阶段，使用一个parser network填充这些void pixels，同时平滑已有的语义分割结果。

一些细节：

(1)在投影到bev的过程中，总是选择高度值更小的点的语义label。

(2)从仿真数据到真实数据迁移时，只用真实数据finetune阶段一的model，阶段二的model保持保持不变。个人理解，这么做的主要原因还是因为bev semantic segmentation的ground-truth相对而言更加难以获取。

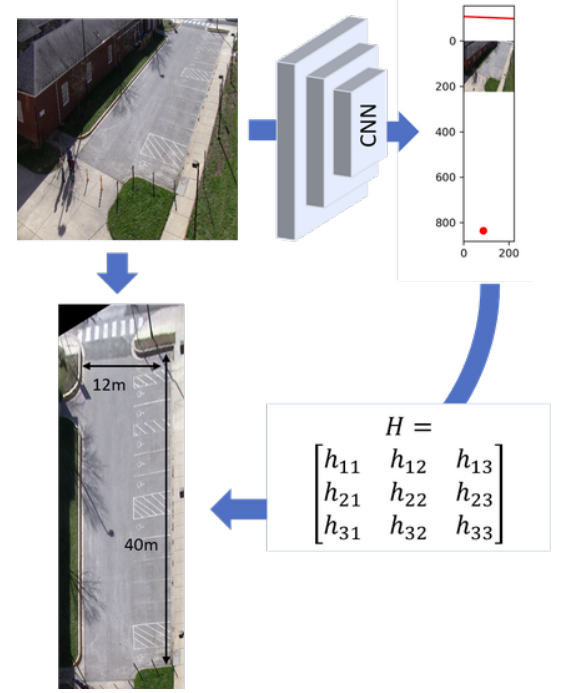


**5. A Geometric Approach to Obtain a Bird’s Eye View From an Image (** [**https://arxiv.org/pdf/1905.02231v2.pdf**](https://arxiv.org/pdf/1905.02231v2.pdf) **)**

这篇文章粗略看了一下，我们知道单应性矩阵具有8个自由度，因此通过4对不共线的匹配点建立线性方程组就可以求解，这篇文章就是这个思路。只不过作者经过分析发现，提出在通常情况下只需要4个参数（对应vertical vanishing point和horizon line）就可以确定这个单应性矩阵，并且作者利用standard stereographic projection对vertical vanishing point和horizon line进行参数重构，以便CNN能够更好地预测它们。

其实这方面的文章也有不少，关于实际应用中单应性矩阵计算方法的总结

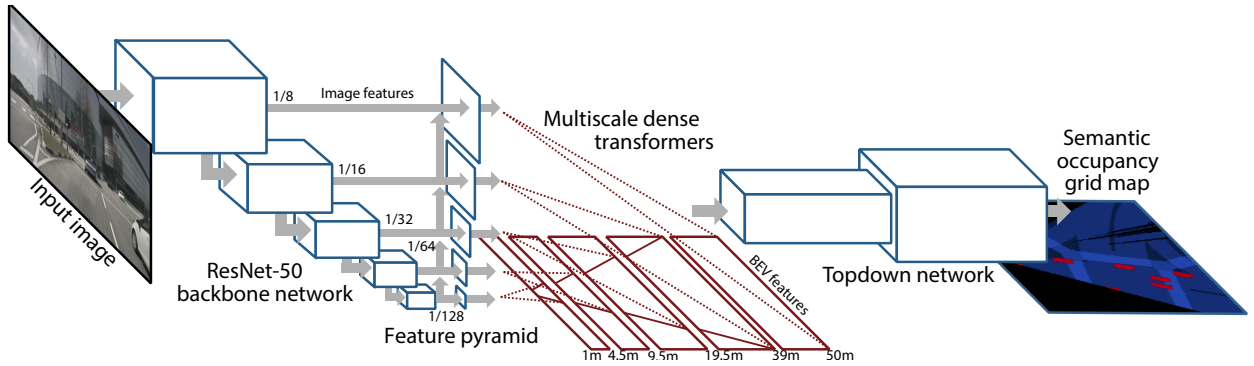
https://blog.csdn.net/feiyang\_luo/article/details/103555036



**6. PyrOccNet(** [**https://arxiv.org/pdf/2003.13402.pdf**](https://arxiv.org/pdf/2003.13402.pdf) **)**

整个pipeline如下图所示，

(1) A ResNet-50 backbone network extracts image features at multiple resolutions. (2) A feature pyramid augments the high-resolution features with spatial context from lower pyramid layers. (3) A stack of dense transformer layers map the image-based features into the birds-eye-view. (4) The topdown network processes the birds-eyeview features and predicts the final semantic occupancy probabilities.



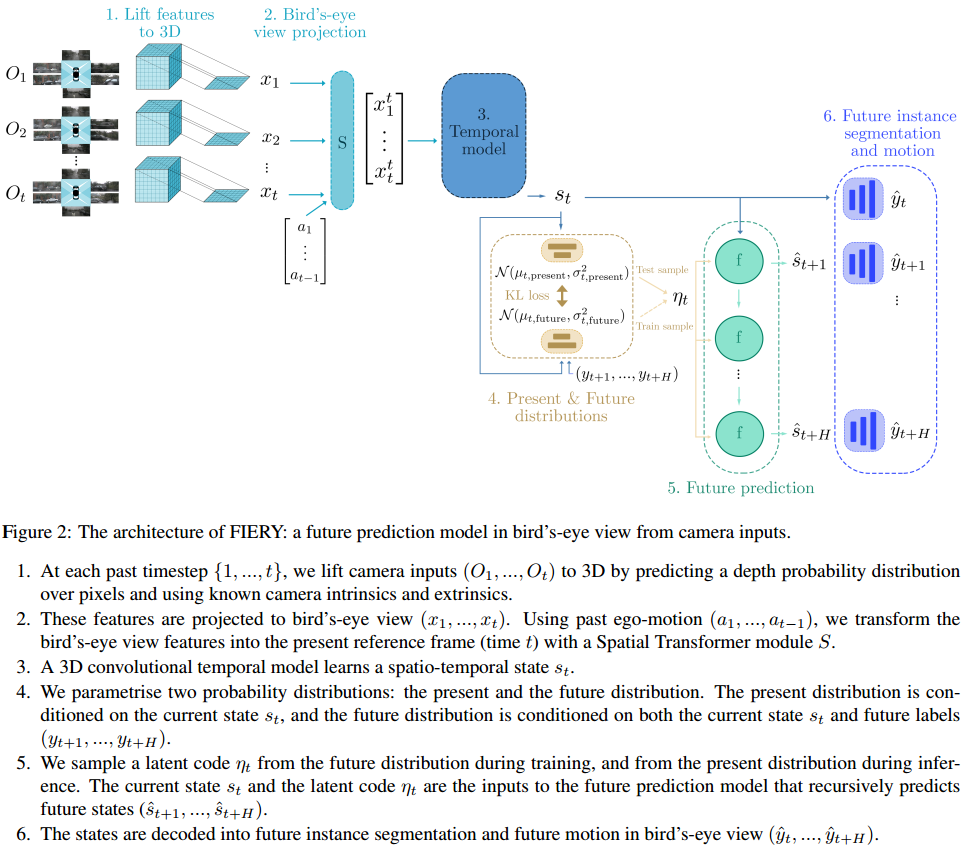
Dense transformer layer

**7. Lift, Splat, Shoot: Encoding Images from Arbitrary Camera Rigs by Implicitly Unprojecting to 3D (** [**https://arxiv.org/pdf/2008.05711.pdf**](https://arxiv.org/pdf/2008.05711.pdf) **)**

**8. Enabling spatio-temporal aggregation in Birds-Eye-View Vehicle Estimation (** [**http://www.computing.surrey.ac.uk/personal/st/R.Bowden/publications/2021/Saha\_ICRA2021pp.pdf**](http://www.computing.surrey.ac.uk/personal/st/R.Bowden/publications/2021/Saha_ICRA2021pp.pdf) **)**

**9. FIERY(** [**https://arxiv.org/pdf/2104.10490.pdf**](https://arxiv.org/pdf/2104.10490.pdf) **)**

整个pipeline如下图所示，

在Lift-Splat-Shoot的基础上加入了temporal fusion，通过融合，比较奇怪的是，从作者的实验结果中发现，