Progressive LiDAR Adaptation for Road Detection

将lidar信息引入视觉道路检测有困难，lidar数据和提取到的特征 和 图像数据和特征不在一个空间上。
论文主要解决了如何将雷达信息引入基于视觉的道路检测。

data space adaptation : transforms lidar data to the visual data space to align with the perspective view by applying altitude(高度) difference-based transformation.

feature space adaptation : adapts lidar features to visual features through a cascaded fusion structure.

目前利用lidar数据提升视觉道路检测的效果不明显。
作者调查了一下为什么不明显，提出了新的方法。
两种困难：

1.lidar 和 camera是两种不同的数据，难以定义一个空间去结合这两种数据。尽管可以将点云投影到图像中，但这回改变道路在lidar 数据中的样子, 使得道路在点云空间中更不易于区分，这会导致深度学习模型无法从lidar数据中学到东西。

2. 难以结合从两种数据提取出来的特征。图像中道路是使用rgb的像素表示，点云中道路是使用离散的点表示，非常可能两者提取出来的特征也在不同空间。

融合视觉和雷达的方法准确率不如纯视觉。

作者提出了一种转换关系能够把雷达数据转换到视觉空间，雷达特征到视觉特征空间。

数据转换的同时能够使点云中的道路容易区分。
然后通过一个串联的融合结构，特征空间转换使得雷达特征更好的补充和提升视觉特征。

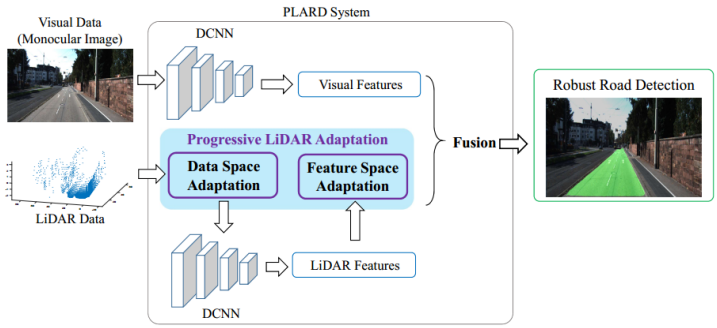
loss:
$\min_W \sum_i \sum_{x,y} \mathcal{L}(f(I_i, L_i; W), \hat{y})|_{x,y}$

i 代表哪个样本

$f(I, L; W) = f_{\text{parsing}}(f_{\text{fuse}}(f_{\text{vis}}(I; W_{vis}), g(L; W_{\text{lidar}})))$
g代表progressive lidar adaptation function
f_{vis} 代表 visual image-based road detection function， ResNet101
f_{fuse} 融合操作
$f_{parsing}$ 最后的二分类, pyramid scene parsing module 后 接 2-class softmax.

g由两部分组成： data space adaptation step and feature space adaptation step.
data space adaptation 中把雷达数据转到2d同时使道路易于区分。
feature space adaptation 中 引入一个可学习的module 将lidar feature 转换到一个更好的补充视觉特征的空间。

$$g\left(L;W_{\text{lidar}}\right)=g_{\text{feat}}\left(f_{\text{lidar}}\left(g_{\text{lata}}(L);W_{\text{lidar}}\right)\right)$$

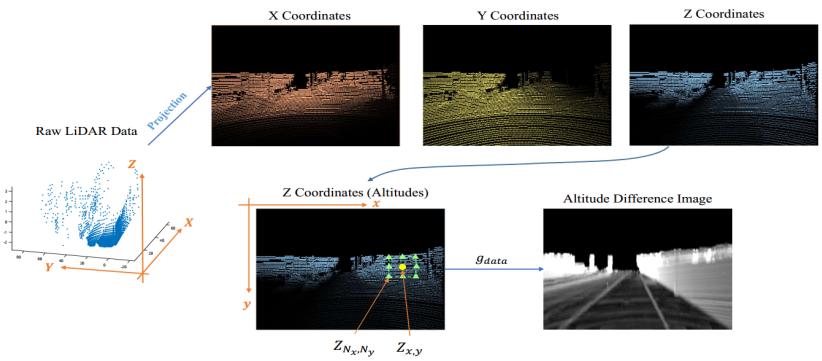


data space adaptation step: altitude difference-based transformation method to transform the lidar data space.

altitude difference-based transformation基于观察，3d空间中的道路表面是平坦的并且在高度防线相对平滑，相对于车辆和建筑来说.
在将点云映射到图像平面后，这种平滑性可以被保存通过记录原始3d点云的高度信息。作为结果，道路区域可以更好的被区分在投影后的点云数据中根据在图像平面中的高度变化。
altitude difference-based transfromation 根据以下公式计算点的坐标:

$$g_{data}(L)|_{x,y}=V_{x,y}=\frac{1}{M}\sum_{N_x,N_y}\frac{\left|Z_{x,y}-Z_{N_x,N_y}\right|}{\sqrt{(N_x-x)^2+(N_y-y)^2}}$$

在图像平面跟周围点做了一个高度差平均。



feature space adaptation:

目的：将lidar feature space 转换到另一个space 去使得lidar feature更好的补充图像特征并提高基于图像的道路检测性能。因为不知道什么样的变换好，所以引入了learning-base module去学习这个操作。

假设线性变换可以定义这个操作：

$g_{\text{feat}}\left(\mathbf{f}_{\text{lidar}}\right)=\alpha\mathbf{f}_{\text{lidar}}+\beta$ ， 缩放加平移

$\mathbf{f}_{\text{lidar}}=f_{\text{lidar}}\left(g_{\text{data}}(L);W_{\text{lidar}}\right)$

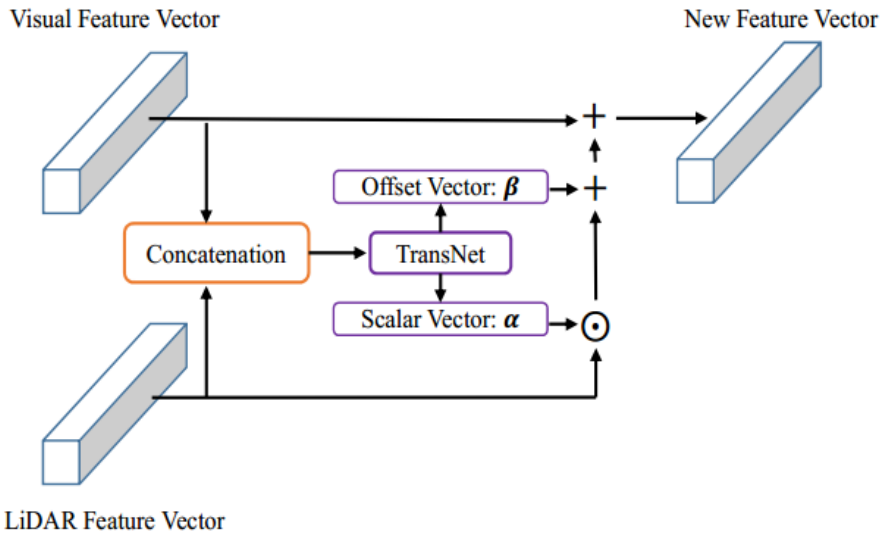
通过神经网络来估计这个变换，

$\alpha=f_{\alpha}\left(\mathbf{f}_{\text{lidar}},\mathbf{f}_{vis};W_{\alpha}\right)$ ， f_{α} 代表计算 α 的神经网络函数， W_{α} 代表对应的权重

$\beta=f_{\beta}\left(\mathbf{f}_{\text{lidar}},\mathbf{f}_{vis};W_{\beta}\right)$

f_{α} 与 f_{β} 都使用全卷积操作。

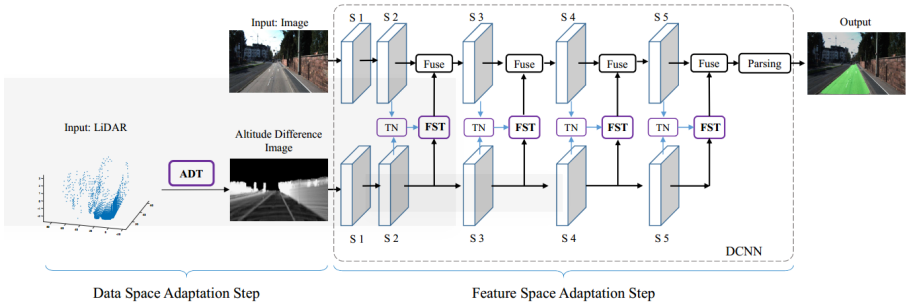
f_{lidar}与 f_{vis} concatenate 作为 f_{α}与 f_{β} 的输入



feature space adaptation 只 包含 3 个 1×1的卷积操作，三个element-wise multiplication or addition operations.

Cascaded Fusion for Adapted Lidar Information

f_{fuse} 通过使用residual-based cascaded fusion structure.



TN代表 transformation network. FST: feature space transformation

$$f_{\text{fuse}}^{k}\left(f_{vise}^{k}\left(f_{\text{fuse}}^{k-1}(I, L); W_{vis}^{k}\right), g^{k}\left(L; W_{lidar}\right)\right)$$
$$= f_{vis}^{k}\left(f_{\text{fuse}}^{k-1}(I, L); W_{vis}^{k}\right) + \lambda g^{k}\left(L; W_{lidar}\right)$$

这里的fuse看起来只是做一个加法,但是通道数量在实现里，lidar缩小了8倍，怎么能直接相加？

fuse的输入是经过变换特征后的lidar feature 和 上一层在fuse后的结果上做卷积的feature,是为了让lidar feature配合visual feature

LOSS:

$$\mathcal{L} = w_{\text{parsing}}\mathcal{L}_{\text{parsing}} + w_{\text{lidar}}\mathcal{L}_{\text{lidar}} + w_{\text{aux}}\mathcal{L}_{aux}$$
三部分组成，三个系数

$$\mathcal{L}_{\{\text{parsing,lidar,aux}\}} = -\sum_{c=1}^{2}\left(\hat{y}^{c}\log_{10}\left(y_{\{parsing,lidar,aux\}}\right)\right)$$

$$f_{\text{parsing}}\left(f_{\text{fuse}}\left(f_{vis}\left(I; W_{vis}\right), g\left(L; W_{lidar}\right)\right)\right)$$
$$\mathbf{f}_{\text{lidar}} = f_{\text{lidar}}\left(g_{\text{data}}(L); W_{\text{lidar}}\right)$$

aux: $f_{\text{fuse}}^{k}\left(f_{vise}^{k}\left(f_{\text{fuse}}^{k-1}(I, L); W_{vis}^{k}\right), g^{k}\left(L; W_{lidar}\right)\right)$
$= f_{vis}^{k}\left(f_{\text{fuse}}^{k-1}(I, L); W_{vis}^{k}\right) + \lambda g^{k}\left(L; W_{lidar}\right)$

lidar 和 parsing 只在最后一个卷积层做
aux只在k=4的卷积层做

实现细节：

7*7 window 处理输入的z

PSPNet as visual image-based DCNN and use ResNet-101 as the backbone

lidar DCNN 每一层channel数比visual 小 8倍， 使用 hybrid convolutions to augment its expressive capacity with fewer channel numbers.

during fusion, we use a uniform channel number ,ie 256(不理解)