# Multi-View 3D Object Detection Network for Autonomous Driving

### 论文地址

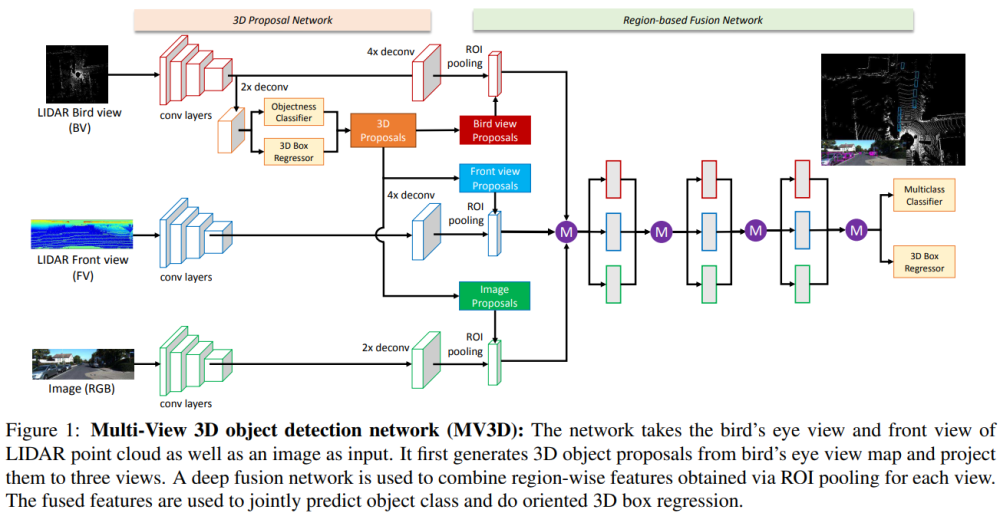
[hhttps://arxiv.org/abs/1611.07759](https://arxiv.org/abs/1611.07759)

### 代码地址

<https://github.com/bostondiditeam/MV3D>

### 模型架构

模型融合了视觉和雷达点云信息，同时与基于voxel的方法不同，MV3D只用了点云的俯视图和前视图，可以降低计算量同时又不至于丧失过多的信息。随后生成3D候选区域，把特征和候选区域融合后输出最终的目标检测框



#### 本文设计的方法具体步骤如下:

* **提取特征**

* + 提取点云俯视图特征

* + 提取点云前视图特征

* + 提取图像特征

* **从点云俯视图特征中计算候选区域**

* **把候选区域分别与1）中a、b、c得到的特征进行整合**

* + **把俯视图候选区域投影到前视图和图像中**

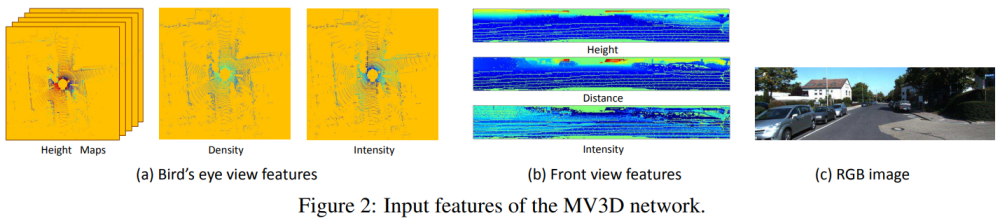
* + **经过ROI pooling整合成同一维度**

* + **把整合后的数据经过网络进行融合**

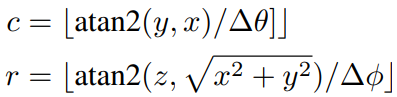
### 提取特征

#### 提取俯视图特征

* 鸟瞰图的表示是**高度、强度和密度编码**的。MV3D将投影点云离散化为分辨率为0.1m的2D网格。对于每个单元，**高度特征计算为单元中点的最大高度**。为了对更详细的高度信息进行编码(**后续基于此可以生成3D propsocal**)，将点云平均分为M个切片。为每个切片计算一个高度图，因此获得M个高度图。强度特征是每个单元中具有最大高度的点的反射率值。点云密度表示每个单元中点的数量。为了规范化特征，将其计算为min(1.0，(log(N+1))/(log(64)))，其中N是单元中的点数。请注意，强度和密度特征是针对整个点云计算的，而高度特征是针对M个切片计算的，因此总的来说，鸟瞰图被编码为**（M + 2）**通道特征



* 前视图表示为鸟瞰视图表示提供补充信息。由于LIDAR点云非常稀疏，因此将其投影到图像平面中会生成稀疏的2D点图。相反，将它投影到一个柱面来生成一个密集的前视图映射。给定3D点p=(x,y,z),可以使用以下公式计算其在前视图中的坐标p\_fv=(r,c), 其中∆θ和∆φ分别是激光束的水平和垂直分辨率



#### RPN网络

* 受区域建议网络(RPN)启发，首先设计了一个网络来生成**三维物体**建议。使用鸟瞰图作为输入。在3D对象检测中，鸟瞰图与正视图/图像平面相比具有多个优势

* + 首先，对象在投影到鸟瞰图时会保留物理尺寸，因此尺寸变化很小，而在前视图/图像平面中则不是这种情况

* + 其次，鸟瞰图中的物体占据了不同的空间，从而避免了遮挡问题

* + 第三，在道路场景中，由于对象通常位于地平面上并且垂直位置的变化很小，因此鸟瞰位置对于获取准确的3D边界框至关重要

* 给出鸟瞰图。网络会从一组3D先验框生成3D框建议。每个3D框都由（x，y，z，l，w，h）参数化，它们是LIDAR坐标系中3D框的中心和大小（以米为单位）。对于每个3D先验框，可以通过离散化（x，y，l，w）来获得相应的鸟瞰锚点（x\_bv，y\_bv，l\_bv，w\_bv）。通过在训练集中聚类地面真实物体大小来设计**N个3D先验框**。**在car detection情况下，先验box的(l,w)取{(3.9,1.6)，(1.0,0.6)}的值，高度h设置为1.56m。通过将鸟瞰图锚点旋转90度，我们得到N = 4个先验框**。(x, y)为鸟瞰图特征点的变化位置，可以根据相机高度和物体高度计算出z。在生成建议时，不做方向回归，而将其留到下一个预测阶段。3D框的方向限于{0◦，90◦}，与大多数道路场景对象的实际方向接近。这种简化使得建议回归的训练更容易

* 离散化分辨率为0.1m，鸟瞰图中的对象框仅占据5-40像素。对于深度网络而言，检测此类超小型物体仍然是一个难题。一种可能的解决方案是使用更高的输入分辨率，但是这将需要更多的计算。这里选择了特征映射上采样。在建议网络的最后一个卷积层之后，使用2倍双线性上采样。在实现中，前端卷积仅进行三个池化操作，即8x下采样。因此，结合2倍的反卷积，馈送至**建议网络的特征图相对于鸟瞰输入降低了4倍的采样率**

* 通过回归到t =(∆x，∆y，∆z，∆l，∆w，∆h)进行3D 框回归。（Δx，Δy，Δz）是通过锚点尺寸归一化的中心偏移量，（Δl，Δw，Δh）由Δs= log s\_GT/s\_anchor ，s∈{l，w，h}计算。使用多任务损失同时分类对象/背景和做3D框回归。特别地，将类熵用于“对象”损失，将Smooth l\_1用于3D框回归损失。计算框回归损失时，将忽略背景锚。在训练期间，**计算锚点和地面真相鸟瞰图框之间的IoU重叠**。如果锚点的重叠度大于0.7，则将其视为正值；如果重叠度小于0.5，则将其视为负值。锚点之间重叠的锚将被忽略

* 由于LIDAR点云的稀疏性，导致许多空锚，因此在训练和测试过程中，会移除所有空锚，以减少计算量。这可以通过计算点占用图上的积分图像来实现

* 对于最后一个卷积特征图每个位置上的每个非空锚，网络会生成一个3D框。为了减少冗余，在鸟瞰框上应用了非最大抑制（NMS）。这里没有使用3D NMS，因为对象在地平面上应该占据不同的空间。对于NMS，将IoU阈值设置为0.7。在训练过程中保留了前2000个框，而在测试中，只使用了300个框

#### 基于区域的融合网络

**多视图ROI池化**

* 由于来自不同视图/模式的特征通常具有不同的分辨率，因此对每个视图采用ROI池化以获得相同长度的特征向量.给定生成的3D proposal，可以将它们投影到3D空间中的任何视图。在示例中，将它们投影到三个视图，即鸟瞰（BV），正视图（FV）和像平面（RGB）。给定一个3D proposal ->p\_3D，通过以下方式获得每个视图的ROI:

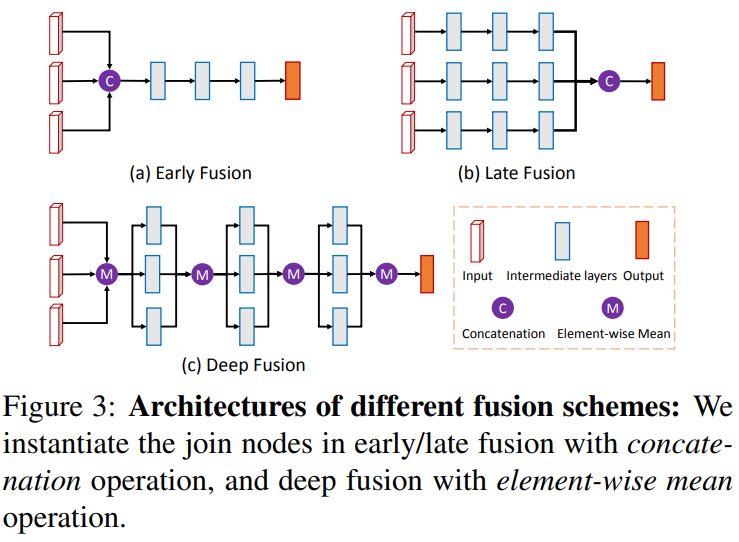


* 其中T3D→v表示从LIDAR坐标系到鸟瞰图，正视图和像平面的转换函数。给定每个视图前端网络的输入特征映射x，通过ROI池获得固定长度的特征f\_v：



**深度融合**

* 尝试多种方法对特征进行深度融合，具体如下图:



#### 实现

**网络体系结构**

在多视图网络中，每个视图具有相同的体系结构。基本网络建立在16层VGG网络上，并进行了以下修改：

* 通道减少到原始网络的一半。

* 为了处理超小对象，使用特征逼近来获得高分辨率的特征图。特别地，在将最后一个卷积特征图送入3D Proposal网络之前，插入一个2x双线性上采样层。同样，在BV/FV/RGB分支的ROI池化层之前插入一个4x/4x/2x上采样层。

* 删除了原始VGG网络中的第4个池化操作，因此网络的卷积部分进行了8倍下采样。

* 在多视图融合网络中，除了原始的fc6和fc7层之外，还添加了一个额外的全连接层fc8。

通过对来自ImageNet上预训练的VGG-16网络的权重进行采样来初始化参数。尽管网络具有三个分支，但参数数量约为VGG-16网络的75％。在GeForce Titan X GPU上，一幅图像的网络推断时间约为0.36s

**输入表示形式**

对于KITTI，它仅提供正视图（大约90°视场）中对象的注释，这里使用[0，70.4]×[-40，40]米范围内的点云。当投影到图像平面时，将删除超出图像边界的点。对于鸟瞰图，离散化分辨率设置为0.1m，因此鸟瞰图输入的大小为**704×800**。由于KITTI使用64光束Velodyne激光扫描仪，因此可以获得前视点的**64×512**地图。RGB图像按比例放大，因此最短尺寸为500**训练**

该网络以端到端的方式进行训练。对于每个小批量，使用1张图像并采样128个ROI，大致将25％的ROI保持为正。我们使用SGD对网络进行训练，其学习率为0.001，可进行10万次迭代。然后，将学习率降低至0.0001，然后再训练20K次迭代

#### 实验

**3D目标检测**

