

传感器联合标定与感知融合（融合篇）

一：背景扯皮

上一篇哔哔了传感器联合标定的相关内容，针对场景主要是 Targetless，其中应用到了的一些数据融合的知识点，如图像点云配对等。这一篇就专门来叨叨一下融合的那些事。

所谓融合，就是把多种不同的元素融合在一起，使其和谐相处。那么感知融合就是把各种传感器形成的各类数据融合在一起，从而达到最佳的探测效果。融合可以看做是一套复杂的系统，为啥非要把各种传感器数据融合在一起呢？上一篇已经做了简单的介绍，这里在大概描述一下。

摄像头是 2D 视觉技术，可以用来传达物体的色彩和形状信息，车载智能摄像头可以输出目标的边框、车道线位置、交通灯和标志状态。摄像头成本较低，但其对自然环境的要求较高，如夜间的识别效果会比白天差。

激光雷达使用光探测和测距，生成点云数据，可以用来探测几乎所有的障碍物，夜间工作也基本不受影响。但激光雷达成本较高，且在雾霾下探测效果大大打折。

毫米波雷达可以探测远距离障碍物，几乎不受自然环境的影响。但目前毫米波精细度不够，而且无法对障碍物进行分类。

下图给出了摄像头、激光雷达、毫米波雷达的优缺点。

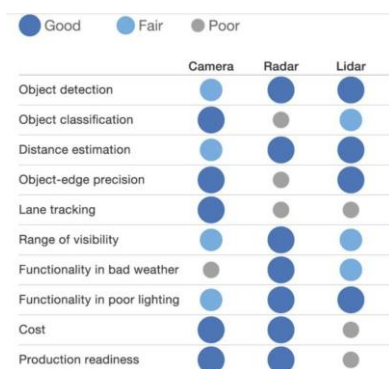


图 1 三种传感器的优缺点

从图 1 可以看出三种传感器之间可以做到互补。就像人的感官系统一样，相互配合才能达到最优。如图 2，车辆运行中，激光和摄像头所获得的目标示意图，感知系统就需要判断激光和摄像头看探测到的是否是同一个物体，这个过程就是融合。

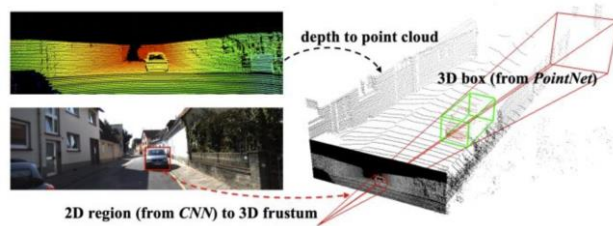


图 2 激光和摄像头所看到的目标的示意图

感知融合按照数据类型可以分为，数据融合、特征融合、目标级融合等，每一块都可以花很多的篇章来描述。对于目标级融合，暂时不做讲解，重点讲解一下针对 Camera 和激光雷达的数据融合与特征融合。看似比较简单了，但其实并非易事，为啥呢？首先 Camera 是通过将物理世界投影到相机平面来记录信息，而点云则将可视区域内物体表面的空间信息，记录并保存。就数据结构和类型而言，点云数据所体现的空间信息是无序的、空间信息与其整体属性信息的弱关联、形态变化丰富；而图像是规则的，局部信息与其整体关联度较大、离散的。这就导致了图像和点云处理算法方面存在较大的差异。



▲微信扫码可查看、购买、学习课程

二、技术扯皮

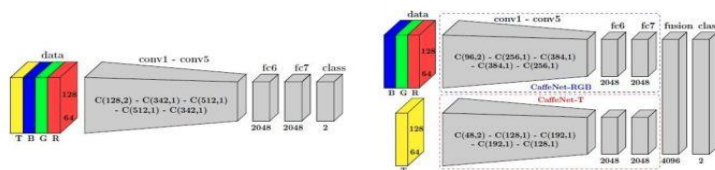


图 3 数据融合与特征融合

扯入正题。上图给出了一个数据融合和特征融合的示意图。左边是数据融合的示意图。首先对二维三维数据简单处理后进行结合，然后通过 CNN 技术，对结合后的数据进行特征提取，直接给出推算结果。这种方法计算成本比较低，但模型缺乏灵活性，而且对传感器标定质量要求比较高。右面是特征融合的示意图，将二维数据和三维数据分别使用 CNN 网络进行特征提取，将特征归一到同一纬度，然后在进行 fusion 层，并给出推算结果。这种算法比较灵活，但算法计算成

本较高，且 fusion 层的设计完全依赖于开发者的经验。

在介绍如何将数据融合前，先大概介绍一下融合的一些必要条件。

1) 运动补偿

所谓运动补偿，就是描述相邻两帧差别的方法。运动补偿包括 ego motion 自身运动补偿和 motion from others 来自其他目标的补偿。

a) Ego motion: 即考虑传感器在采集过程中的某一时间戳内，由于车辆自身的运动，采集的对象会在该时间戳内发生相对位移变化；以激光雷达为例，采集一圈需要 0.1s，在这 0.1s 内，车身本身会发生一定的位移，如果不考虑车辆本身位移的情况，检测出来的目标位置就会产生较大误差；

b) motion from others, 考虑传感器在采集过程中的某一时间戳内，运动物体由于自身运动会产生相对位移变化。

2) 时间同步

时间同步比较好理解，激光雷达和相机具有不同的帧率，且每个传感器都有自己的时间源，另外数据传输、相机曝光等都会产生不可控的延迟。那么时间同步就是解决该误差的一个方法，通过统一的主机给各个传感器提供基准时间，各传感器根据已经校准后的各自时间为各自独立采集的数据加上时间戳信息，以做到所有传感器时间戳同步；但由于各个传感器各自采集周期相互独立，无法保证同一时刻采集相同的信息。通常在车载上，大多数支持 GPS 时间戳的时间同步方法。



▲微信扫码可查看、购买、学习课程

3) 空间同步

所谓的空间同步就是将不同的传感器坐标系测量值转换到同一个坐标系中。
(上一张联合标定好像讲了哈)。

回归正题，接下来先来扯一下如何进行 2D、3D 的数据融合。在之前，通常

将激光点云数据(3D)投影到 2D 图像上，然后进行像素点的关联来实现。这种方法对图像和点云之间的分辨率要求比较高，低分辨率的点云投影到高分辨率的图像上会产生空洞现象，为了解决这一现象，融入一些额外的信息（如深度补全法或者点云上采样等）在某一程度上缓解这一现象。目前比较流行的一种方法就是将图像和点云特征转换为其他数据格式来表示，如 Voxel。

很尴尬，不知所云。放一张图缓解一下局面。

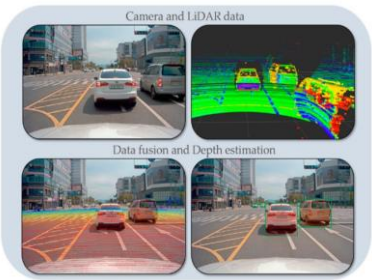
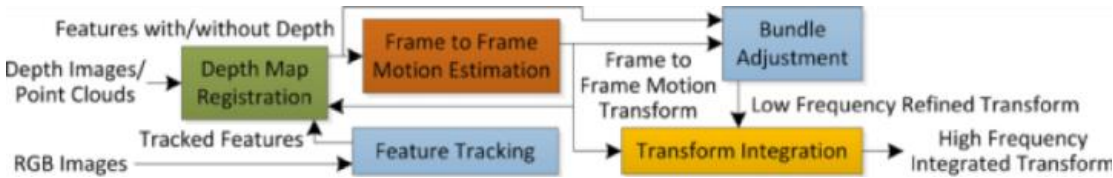


图 4 尴尬缓解图片

不扯，聊技术。先来聊一下数据融合（传统的）方法，然后再说一下特征融合（深度学习）方法。为了实现数据融合，就是要实现图像与深度直接按的结合，这就需要图像特征与深度图特征之间的相关性，更多表现为 pixel(像素)/depel(深度素)/surfel(表面素)/voxel(体素)等。

《Real-time Depth Enhanced Monocular Odometry》是 Ji Zhang 大神 2014 年的一个里程计，该里程计是通过单目相机获取 RGB 图像，通过 Lidar 或者 RGBD 相机获得深度信息，然后对 RGB 图像进行特征点的提取，然后对深度图进行处理，将相邻两帧图像能够匹配上的特征点所对应的深度值进行分类，主要分为传感器获取的，三角化、kd-tree 恢复的，以及无法获取深度信息的。然后根据分类利用不同的公式，进行位姿求解；最后进行 BA 优化（每 5 帧选一帧）。

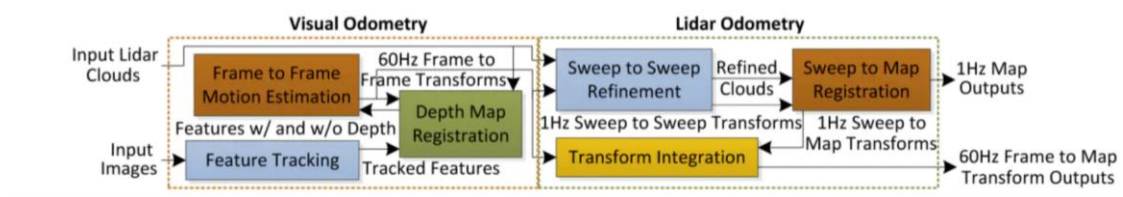
该系统框架如下：



上面的架构图，是不是就是 VIO 上加了一个激光雷达点云信息。上图中，feature tracking 使用 opencv 稀疏光流实现特征跟踪，为了使特征分布均匀，将图像划分为很多个 cells，每个 cell 都尽可能提取特征点(VIO 的前端追踪算法)；depth map registration 负责将局部地图投影到前一帧用于恢复前一帧中没有恢复

深度的特征点赋予深度信息；Frame to Frame Motion Estimation，主要利用深度的特征点提供的重投影误差(点到点的距离)和没有深度特征点提供的基础矩阵约束(点到极线的距离),来求解帧间位姿；Bundel Adujstment 类似于滑窗优化，细优化位姿。

《Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast》也是一篇在 VLOAM 的经典工作。它首先用 VO 估计去匹配激光点云，修正高频点云的畸变，然后基于扫描匹配的激光雷达测距仪同时改进运动估计和点云配准，随后逐步实现地图。



与上一篇相比，该系统也采用了 VO 估计，VO 框架除了帧率外，几乎通用；而不一样的是 Lidar Odometry。该系统采用 VO 和 LO 两种方法的结合，充分利用了每个传感器的优点，并补偿了另一个传感器的缺点。还能够消除由于视觉测距的漂移引来的点云中的失真，然后利用修正模块来构建相对于地图的传感器姿态。

上面两篇均是基于 LOAM 的融合方法，LOAM 融合方法可以将 RGB 图和深度信息充分结合，然后逐步优化传感器姿态，但该方法依赖于建图，运行速度较慢。这里再介绍一篇基于传统滤波的文章《In Defense of Classical Image Processing: Fast Depth Completion on the CPU》。

该文章发表于深度学习横行的 2018 年，居然没有依赖任何训练数据，也没有使用深度学习模型，仅仅使用了经典的图像处理算法进行深度补全算法。这也是介绍本文的原因。



▲微信扫码可查看、购买、学习课程

该方法共有 8 个步骤：

1) **Depth Inversion**: 对于稀疏图像的处理机制，形态学处理方法一般是使用较大的像素值覆盖较小的像素值。对于普通的相机，其深度值是估算出来的，假设深度值为 0~80m，对于没有有效深度信息的地方，会用 0 值来填充深度值。如果这时采用形态学来处理，会直接使得景深距离覆盖掉小距离，从而导致较近的边缘信息丢失。而 **Depth Inversion** 就是将有效深度做一个 invert，即 $D(\text{inverted}) = 100 - D(\text{input})$ ，这样在有效深度和空值中间有了一个 20m 的 buffer。

2) **Custom Kernel Dilation**(用自定义的核进行膨胀操作): 先填充最接近有效像素值的空置，关键在于 **kernel** 形状的设计（可以理解为 CNN 卷积核的设定），要使得最可能具有相同深度值的像素被填补。所以这一步采用一个 5*5 的 **kernel**。如下图（确实和 CNN 挺像）：



0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0

3) **Small Hole Closure**(对小的空洞进行闭运算): 经过第一步的膨胀操作后，仍存在一些小的孔，这些区域都没有有效的深度值，考虑到物体的结构，发现膨胀后相邻的深度可以连接起来形成物体的边缘。所以这一步用一个 5*5 的 **full kernel** 进行闭运算，连接相邻的深度值，填补小孔。



Full				
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

4) **Small Hole Fill**(小孔填充): 经过前面的步骤，深度图中仍然有一些小中 size 的洞没有被填充，为了填补这些洞，首先计算一个空像素的 **mask**，然后再进行 7*7 的全内核膨胀操作。这个操作只会填补空像素，同时保持先前计算过的有效像素不变。

5) **Extension to Top of frame**（延伸到顶部）: 考虑一些比较高的物体，如树木建筑物等，都到达了 **LiDAR** 点云的顶端。为了补齐这些物体的深度，将每一列的

顶端的像素值向上延伸。

6) **Large Hole Fill (大洞填充)**: 这一步来填充一些比较大的洞。用一个 31×31 的 **full kernel** 来进行膨胀操作, 将所有仍然为空值的深度值填充起来, 而其他位置的有效深度值不变。

7) **Median and Gaussian Blur (中值/高斯模糊)**: 前面步骤中的膨胀操作会引入一些噪声, 为了去除这些噪声, 先使用 5×5 **kernel** 的中值模糊, 去除局部边缘的离群点。然后, 再用 5×5 的 **kernel** 进行高斯模糊, 对局部平面进行平滑, 平滑尖锐的物体边缘。

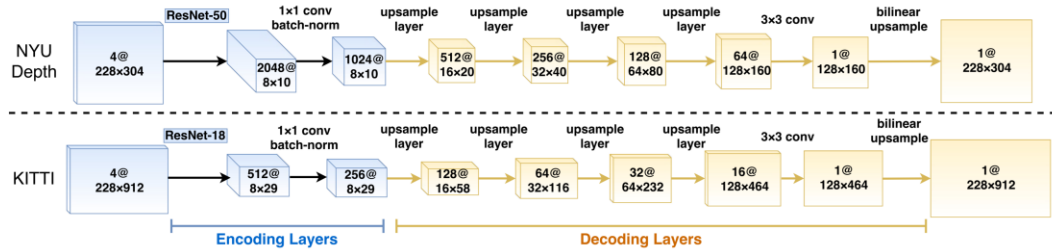
8) **Depth Inversion**: 算法的最后一步是将前几步所使用的倒置深度值恢复到原来的深度编码, 即 $D(\text{inverted}) = 100.0 - D(\text{input})$ 。

可以看到, 该算法采用了一种贪婪策略, 有效深度周围的空值可能具有相似值, 所以采用了先填充小 **hole**, 再填充大 **hole** 的顺序, 最后减少输出噪声和平滑局部平面。其核心在于 **kernel** 形状和大小的设计, 如果太大, 填充的深度值大于实际作用区域, 太小, 那么膨胀操作时, 不足以使边缘通过后期的闭运算连接起来。

那么, 问题来了, 真的有用吗? 根据上述的表征, 再稀疏度比较均匀的图像中, 上述方法能够取得不错的效果。对于自动驾驶, 如果只关注路面上的障碍物, 那么图像上半部分几乎都是空置, 其效果一般般。不过上述思路倒是可以借鉴的。

不扯了, 直接扯一下基于深度学习的特征融合技术。基于深度学习的特征融合技术有很多文章, 其输入比其他深度学习输入多了一个 **mask** 图 (其实就激光雷达投影到图像平面有值的标志图)。这里介绍几篇比较经典的:

《**Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image**》这篇文章设计了 **Sparsity Invariant CNN** 用于深度图像预测的深度回归模型。该模型的输出除了 **RGB** 图像外, 还有一组稀疏的深度测量值, 来进行全分辨率的深度重建。这种稀疏深度测量值可以依赖低分辨率深度传感器或者 **SLAM** 方式来获得。先上网络结构。



网络结构由 Backbone、Encoding、Decoding 三个部分组成。这里 Backbone 使用的 ResNet 系列，Encoding 层由 3*3 的卷积层和 1*1 的卷积+BN 层构成，然后解码层是由 4 个上采样层和 1 个双线性上采样层组成。最后在使用一个 bilinear upsampling 的上采样层来达到一个比较好的精度。

这篇文章中另外一个比较有意思的就是根据真值创建输入稀疏深度图像的采样策略。在训练阶段，文章动态的从 GT 深度图像 D^* 中随机采样输入的稀疏深度 D 。对于任何深度样本的目标数量 m ，有伯努利概率 $p = m / n$ ，其中 n 表示 D^* 中有效深度像素的总数。那么对于任何的像素，则有：

$$D(i, j) = \begin{cases} D^*(i, j), & \text{with probability } p \\ 0, & \text{otherwise} \end{cases}$$

通过这种采样策略，每个训练样本的非零深度像素的实际数量就会在期望的 m 值左右变化（比如深度为 $2m$ 的点占比为 1%，那么概率 p 为 0.01，那么对应 0.01 概率的点就被保留，其他置零）。这种采样目的，就是针对不同数量的输入来提高网络的鲁棒性，并创建更多的训练数据。

完事，看一下效果。貌似还不错。

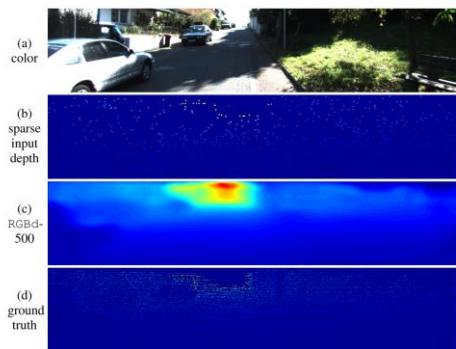


Fig. 4: Example of prediction on KITTI. From top to bottom: (a) RGB; (b) sparse depth; (c) RGBd dense prediction; (d) ground truth depth projected from LiDAR.

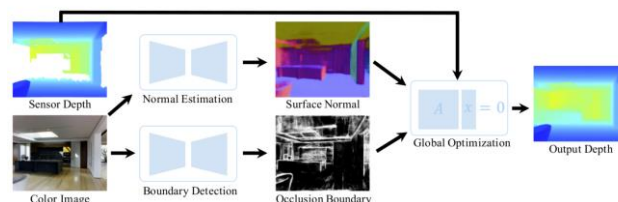
《Deep Depth Completion of a RGB-D Image》这篇文章用在 3D 重建和深度估计上比较多，很少用在融合上。但是放在这里，也是因为它在进行 RGB 图像

与 Depth 图进行任意形式深度图补全时，采用的结果消融和跨数据集的研究。
对 Camera 和 Lidar 的融合有很好的帮助。



▲微信扫码可查看、购买、学习课程

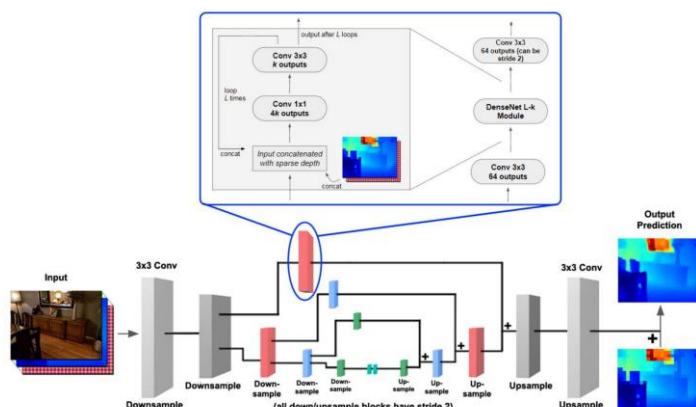
废话不多，先上图。这张图其实和这篇文章没啥关系，但架构很类似。



只不过这篇文章略微做了一些修改，其中 color 图部分，其强度值归一化为 $[-0.5, 0.5]$ ，然后使用双线性插值来调整彩色图像的大小。Depth 图，以 m 为单位的深度绝对值来作为输入，假设没有深度信息，那么像素被赋值为 0。为了解决没有深度和距离 0m 之间的关系，特意添加了一个二进制 mask 来进行深度标记。所以深度输入包含两个通道（深度值和二进制掩码）。然后使用最近邻搜索来调整深度图的大小，以消除深度消失带来的不准确平滑。

这里的 Loss 采用的时 smooth L1 loss 用于深度训练和求导，采用 cosine embedding loss 来训练表面发现，使用交叉熵损失，来进行遮挡边界检测的分类。

OK，这篇文章似乎也就没啥了。有点像单目深度估计的思路。下面这篇比较搞笑，叫 D3 网络，全称 Deep Depth Densification，文章名为《Estimating Depth from RGB and Sparse Sensing》。相信单目深度的小伙伴也都看过这篇文章。其思路也是将 RGB 图、深度图和 Mask 图输入生成两个特征图，然后将两个特征图融合为一个 feature map。废话不多说，直接上图。



该文章对输入参数进行了参数化处理，这种处理可以在任何时候，对任意稀疏深度的输入均适应。这里参数化的输入包括：

1. RGB 矢量值图像 I 以及地面真实深度 D 。这两个地图的尺寸均为 $H \times W$ 。 D 中无效值被编码为 0。
2. 二进制图案掩码 $mask, M(x, y)$ ，其尺寸为 $H \times W$ 。 $M(x, y)=1$ 定义了所需要的深度样本 (x, y) 位置。对于 $M(x, y)$ 进行预处理，以便 $M(x, y)=1$ 的所有点都必须对应于有效深度点。

从上面的 I, D, M ，可以为稀疏深度输入形成两个映射图(map)：

$S_1(x, y)$ ：稀疏深度 $M(x, y) * D(x, y)$ 的 NN(最近邻)填充。

$S_2(x, y)$ ： $M(x, y)$ 的欧几里得距离变换，即 (x, y) 与最近点 (x', y') 之间的 L2 距离，其中 $M(x', y')=1$ 。

两个映射图的尺寸均为 $H \times W$ 。

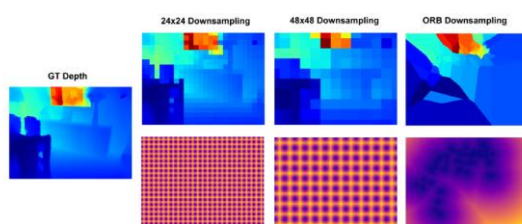


Fig. 2: Various Sparse Patterns. NN fill maps S_1 (top row) and the sampling pattern Euclidean Distance transforms S_2 (bottom row) are shown for both regular and irregular sparse patterns. Dark points in S_2 correspond to the pixels where we have access to depth information.

OK，没啥新东西了。接下来就是将 S_1 、 S_2 串联的 RGB 图像作为输入，然后第一个和最后一个计算块为简单的 3×3 stride-2 卷积，其他所有块均为 DenseNet 模块。网络中所有的卷积层都进行 bn 操作，并接入 ReLu。最后输出一个残差，该残差被添加到稀疏深度图 S_1 ，以产生最终的密度深度预测。

直接贴上 D3 的算法原理吧。

Algorithm 1 Sparse Inputs for the Deep Depth Densification (D³) Model

```

INPUT image  $I(x, y)$ , depth  $D(x, y)$ , and pattern mask  $M(x, y)$ .
INITIALIZE  $S_1(x, y) = 0$ ,  $S_2(x, y) = 0$  for all  $(x, y)$ .
FOR  $\mathbf{r} := (x, y)$  s.t.  $D(\mathbf{r}) = 0$  AND  $M(\mathbf{r}) = 1$ :
     $\mathbf{r}_{\text{new}} = \text{argmin}_{\mathbf{r}'} \|\mathbf{r} - \mathbf{r}'\|_2$  s.t.  $D(\mathbf{r}') > 0$ ;           ( $\|\cdot\|_2$  denotes the  $L_2$  norm.)
     $M(\mathbf{r}) = 0$ ;            $M(\mathbf{r}_{\text{new}}) = 1$ ;
ENDFOR           (All depth locations are now valid.)
FOR  $\mathbf{r} := (x, y)$ :
     $\mathbf{r}_{\text{nearest}} = \text{argmin}_{\mathbf{r}'} \|\mathbf{r}' - \mathbf{r}\|_2$  s.t.  $M(\mathbf{r}') = 1$ ;
     $S_1(x, y) = D(\mathbf{r}_{\text{nearest}})$ ;            $S_2(x, y) = \sqrt{\|\mathbf{r}_{\text{nearest}} - \mathbf{r}\|_2}$ ;
ENDFOR
OUTPUT concatenate( $S_1, S_2$ )

```

好了，上面似乎也和融合没太多关系，但其实深度图的生成，一般依赖于激光去扫描。所以，可以看作 Lidar 和 Camera 的一种融合方式。下面就介绍一篇用于 Lidar 和 Stereo 融合的文章，看完后就会发现和上面提到的几篇其实一样一样。

《High-precision Depth Estimation Using Uncalibrated LiDAR and Stereo Fusion》，这篇文章没有事前对传感器进行校准。

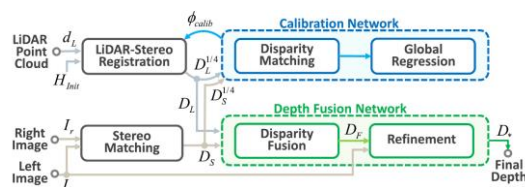


Fig. 2. Flowchart of proposed system. Our system takes the LiDAR point cloud and stereo image pair as inputs and produces the high-precision disparity as outputs.

按照惯例，首先上结构图。这篇文章的主题思想其实就是在合并 RGB 图像和 depth map 之前，先通过几个 convolution layer 提取 feature map。下面来详细介绍一下。从上图可以看出，该结构由校准网络 Calibration Network 和深度融合网络组成。这两个网络的设计都考虑了传感器的精度和效率之间的权重。CalibrationNetwork 首先对初始外部参数进行矫正，通过在深度域中进行标定，可以提高标定的精度（以，又来标定了）。在深度融合网络中，对稀疏 Lidar 和密集立体深度的互补特征进行 boosting 编码。

OK, CalibrationNetwork 的结构图如下，它首先是将 Lidar 得到的点云通过一个初始的但不精确的外参矩阵 \mathbf{H} 投影到双目相机的左侧图上（初始外参精确度越好，收敛越快）。

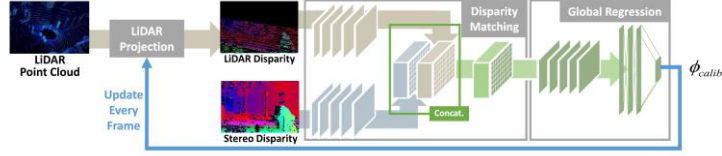


Fig. 3. Illustration of the calibration network. Calibration network takes LiDAR and stereo disparities as inputs and produces calibration parameters as outputs. The proposed network consists of disparity matching and global regression modules. LiDAR and stereo disparities are registered based on calibration parameters updated every frame.

$$[u, v, 1]^T = P H_{init} \phi_{calib}^{-1} [x, y, z, 1],$$

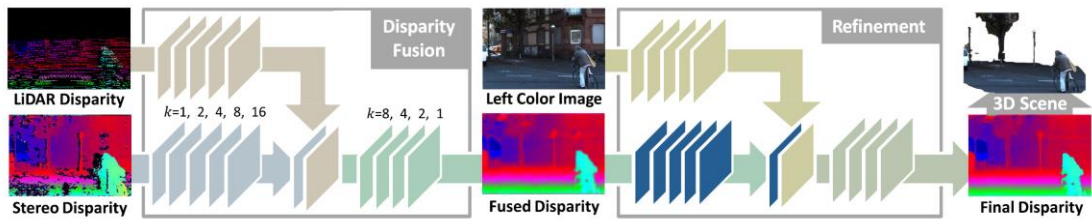
这里 $[x,y,z,1]$ 是由 LIDAR 得到的三维点云的坐标。 $[u,v,1]$ 是图像坐标系， P 是内参矩阵， ϕ_{calib} 是用来校准初始外参矩阵，其表达形式为：

$$\phi_{calib} = \begin{bmatrix} \mathcal{R}(r_x, r_y, r_z) & [t_x, t_y, t_z]^T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

这里的 R, t 就是传感器标定的旋转、平移矩阵。 ϕ_{calib} 会随着每一帧迭代而更新。文章先将 Lidar 点云的深度信息通过 $D_L(u, v) = b_s f_u / x$ 转换为视差值。这样就可以将原来的深度和视差的 different modality data 问题转换为 disparity domain 来解决了。

视差匹配模块，从 $D_L^{1/4}$ 到 $D_s^{1/4}$ 提取中间特征，然后分别从 D_L 和 D_s 向下采样，缩放因子为 4，通过级联和卷积层来估计它们之间的特征对应关系。这里采用下采样是为了在保证高精度的同时，对 highly textured surfaces and shadows 更加鲁棒，同时对于相机模型的变化也会更加鲁棒。校准网络每帧都会更新校准参数，且输入的视差图会在 Lidar 投影模块中重新调整为原始图像分辨率。

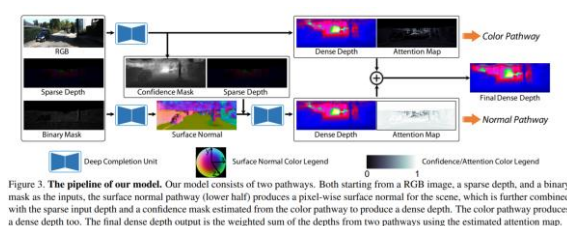
深度融合网络如下图，是由两个级联子模块组成，包括视差融合和优化。其融合模块使用了膨胀卷积操作和 UNet 结构，而优化模块使用一个残差网络来进行深度图预测。这种设计的原因有 1) 3D 激光视差和双目视差可以提供互补的线索，以帮助重构高精度视差；2) RGB 图像可以用来提高视差估计性能。



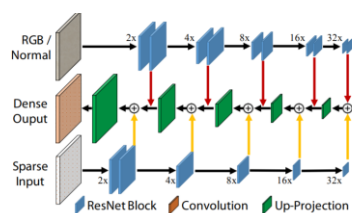
OK，效果咋样，别问我，自己去实验。接下来介绍最后一篇融合的文章，叫《DeepLidar: Deep Surface Normal Guided Depth Prediction for Outdoor Scene from

《Sparse Lidar Data and Single Color Image》名字好长，这篇文章除了上面提到的 Mask 图外，还引入了 surface normal 图来增强 depth prediction，同时还特别的引入了 attention 机制来做目标驱动。

这篇文章可以认为是前面几篇的集合，其即使用了曲面估计方法（和上一篇的标定模块很像），也应用了编解码架构。同时还将前文提到的室内场景搬到了室外。这样做既可以产生稠密深度，还可以高效融合彩色图像与稀疏的雷达深度信息，为了应对场景的挑战，还预测一个置信度掩膜，以处理由于遮挡造成的前景边界附近混合的激光雷达信号。



从上面网络图可以看出也分为两个彩色图分支和曲面法线分支。前者把激光雷达重新投影的彩色图和稀疏深度作为输入，并输出一个完整的深度；后者首先从输入的彩色图和稀疏深度中预测一个曲面法线图像，然后与稀疏深度和置信度掩膜相互整合，从而生成一个完整的深度。这两条路径，都有一个深度补全单元 DCU 来实现，然后将这两个路径通过一个学习好的加权和来整合以产生最终的完整深度。



这张图看的比较清晰。RGB/normal 和稀疏深度图首先通过一个编码器（这里主要是一系列的 ResNet 模块）学习到 local affinity，然后利用解码器（包含了 4 个 up-projection 单元）来逐步增加特征分辨率，对来自另一个编码器输入的稀疏深度特征进行插值，从而给出稠密的输出。

这篇文章用到了注意力机制，即利用来自彩图的先验生成噪声敏感区域的深度，同时结合自有估计出的曲面法线深度来进行对比，得分高的，就会被作为该区域的最终深度。

唠唠叨叨了 N 久,有点乱了。还是看文章吧。联合标定和融合部分告一段落。后面有机会再整理吧。对自动驾驶的 Lidar 和 Camera 的融合做一个小的总结。

从工程角度来讲,自动驾驶的冗余设计对其安全至关重要。尽管 Lidar 与 Camera 的融合可以改善感知性能,但也带来信号耦合的问题。如果在工作时有一条信号路径失效,那么整个流程可能都会发生故障,影响下游模块。这对自动驾驶来说是不可接受的。这一问题可以通过加入能够接受不同传感器输入的多个融合模块或者异步多模数据融合模块来解决,但性能受一定的影响。另外自动驾驶必然面临着对抗攻击(针对基于图像的感知系统的对抗对自动驾驶来说已经构成了严重的危险)在这种情况下,可以进一步探索如何利用 Lidar 的准确信息和图像共同对抗这些攻击。另外,环境中也常常会遇到极端案例,在考虑融合算法时,对于不常见的特殊障碍物或者极端的天气,应当利用多模数据来识别这些极端情况,可能会比用单模传感器更有效、可靠和简单。



▲微信扫码可查看、购买、学习课程