doi:10.6041/j.issn.1000-1298.2019.S0.050

基于双目视觉的空间语义网络实现算法

龚章鹏　王国业　彭思杰
(中国农业大学工学院, 北京 100083)

摘要: 环境感知是无人驾驶中的重要环节。针对目前广泛使用的感知设备——激光雷达存在价格昂贵、信息单一的问题,基于深度学习技术,提出一种同时具备图像分割、双目立体估计功能的联合训练网络,即空间语义网络(SSN)。通过空间映射,SSN 可实现输入双目图像、输出语义点云的功能。经 KITTI 数据集训练后,用测试集验证,结果显示,本文模型对图像分割准确率可以达到 82.5%;针对近点,以立体估计误差 5% 以内判定为准确,立体估计准确率可以达到 95.5%。同时,运算速度可以达到 0.135 s/帧,每帧生成约 4.8 万个语义点云坐标,接近低速工况下的实时性要求,具有较强实际应用价值。

关键词: 空间语义网络; 深度学习; 图像分割; 立体估计; 双目图像; 语义点云
中图分类号: TP391.4　　**文献标识码**: A　　**文章编号**: 1000-1298(2019)S0-0324-07

Spatial Semantic Network Implementation Algorithms Based on Binocular Vision

GONG Zhangpeng　WANG Guoye　PENG Sijie
(*College of Engineering*, *China Agricultural University*, *Beijing* 100083, *China*)

Abstract: Environmental perception is a vital part of driverless driving. The present widely used radar is trapped with its expensive cost and unitary information. Based on the deep learning technology, a joint training network referred to as spatial semantic network (SSN) was proposed, which can realize image segmentation and stereo estimation simultaneously. Through spatial mapping, SSN can input binocular images and output semantic point clouds. The SSN was trained by the KITTI dataset, and then the trained model was validated by KITTI test set, of which the verification result showed that the accuracy of image segmentation can reach 82.5%. And for the near points, the accuracy of stereo estimation can reach 95.5%, where the error within 5% was considered as accurate. Moreover, the processing speed can reach 0.135 s per frame, generating around 48 000 semantic cloud point coordinates per frame, which was close to the real-time requirement under low-speed conditions, and had strong practical application value.

Key words: spatial semantic network; deep learning; image segmentation; stereo estimation; binocular images; semantic cloud points

0　引言

环境感知是无人驾驶以及智能设备实现智能控制的一个重要环节。在感知环节,需要知道周围物体的方位信息、属性信息,并转换成有助于计算机理解的数据形式。激光雷达形成的点云位置信息有利于环境空间位置信息的生成,使其在即时定位与建图(Simultaneous localization and mapping, SLAM)和智能控制感知端得到了充分应用[1]。但激光雷达获得的数据形式较为单一,缺少对环节理解有重要意义的语义信息,使其难以单独作为感知部件延伸到除避障功能以外的应用领域,常需要额外

收稿日期: 2019-04-20　修回日期: 2019-05-20
基金项目: 国家自然科学基金项目(51775548)
作者简介: 龚章鹏(1992—),男,博士生,主要从事车辆智能控制研究,E-mail: gzp_cau@126.com
通信作者: 王国业(1963—),男,教授,博士,主要从事车辆智能控制研究,E-mail: wgy1615@126.com

的传感器，比如相机。无人驾驶对安全性极其敏感，冗余十分必要。因此，相机具备激光雷达的功能，相机与激光雷达之间不是简单的功能补充，而是交互冗余验证，这对提高感知端安全性具有重要意义[2]。在农业领域，激光雷达依旧是最理想的感知设备之一[3-4]，考虑到激光雷达设备价格昂贵，目前多采用深度相机取代激光雷达作为感知设备[5-7]。但深度相机精度不高，户外场景下受光线影响大，无法完全取代激光雷达获取精确的点云数据。因此，迫切需要低成本、高性能的感知方案，这也有助无人驾驶技术在农业领域的应用推广。

近年来，大量的研究工作投入到挖掘机器视觉研究中，特别是深度学习技术的应用，使得机器视觉在图像分类、目标识别、语义分割等方面都取得了很好的应用效果[8]。全卷积网络（Fully convolutional networks，FCN）[9]是最为经典的语义分割网络，它利用全卷积将图像分割问题转换为像素点分类问题，利用反卷积[10]还原尺度。瀑布式网络（Image cascade networks，ICNet）[11]是一个接近达到实时处理的语义分割网络，多分支瀑布式的网络结构参考了残差网络[12]的思想，将浅层信息与深层信息融合，在共享浅层网络权值的同时，用上采样代替部分较为耗时的反卷积步骤，大幅减少了运算时间。金字塔立体匹配网络（Pyramid stereo matching network，PSMNet）[13]是目前双目估计最好的网络之一。离散化的方式使得左右双目图像的不一致性得到有效表达。

目前，多数研究都是针对单独的任务设计网络，应对多个任务需求时，多个网络并行计算将消耗大量内存和计算资源，大大增加了计算时间，对硬件设备的要求也更高。多个子任务网络共享浅层权值，形成一个多任务网络模型是一种更合理的方式。文献[14]基于单目相机实现了同时完成目标识别、距离估计和语义分割的多任务网络模型。但目前尚未研究同时完成双目立体估计和语义分割的多任务网络模型。

首先，双目视觉更符合人的感知习惯，后方背景与前方物体必定是两个不同的物体，即双目匹配图像有助于语义分割准确率的提高。但后方物体与前方物体也有可能属于同一类物体，这又会降低语义分割的准确率，需要平衡这两点的影响关系。其次，双目匹配生成的不一致性特征图与语义分割生成的结果同为像素级数据，更适合融合。因此本文提出一种能够同时完成双目立体估计和语义分割任务的联合训练网络——空间语义网络（Stereo sematic network，SSN），以期仅依靠双目摄像头，实现分类点云的输出效果。

1　空间语义网络（SSN）

SSN 由立体估计子网络与分割子网络两部分组成。整体网络结构见图 1。各网络层参数和节点输出尺寸见表 1。

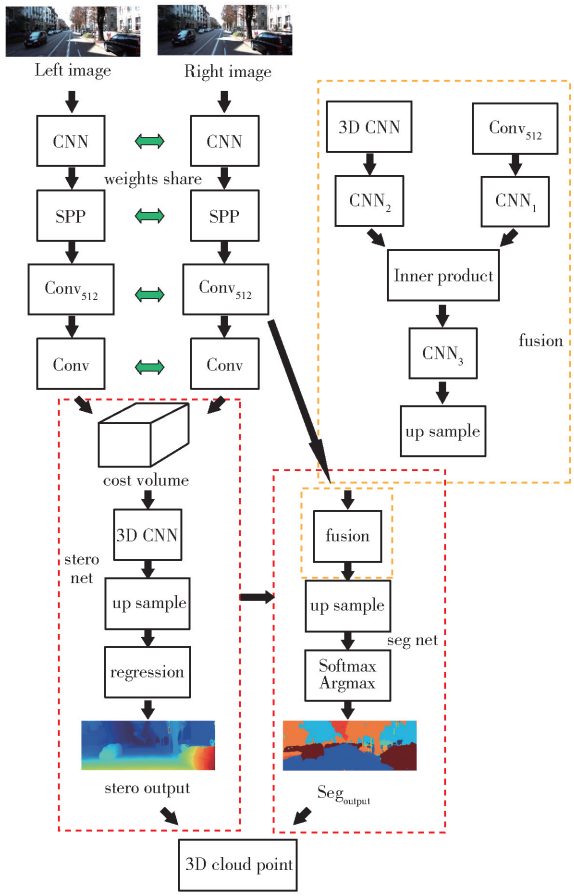

图 1　SSN 模型结构

Fig. 1　Architecture of SSN

1.1　网络整体结构

SSN 语义分割和立体估计两个子网络共用一个特征提取层。左相机图像与右相机图像分别传入特征提取层，得到两个特征图 $Conv_{left}$、$Conv_{right}$。$Conv_{left}$、$Conv_{right}$ 传入立体估计子网络。立体估计沿用了 PSMNet 中的不一致性离散化设计。对 $Conv_{right}$ 左侧补零后，用 $Conv_{left}$ 等大的窗口向左逐像素滑动，取切片，并分别与 $Conv_{left}$ 在特征维度上堆叠。通过三维卷积将堆叠的特征维度消去，每个像素中的每个切片会对应一个特征值，将左图某个像素位置点的特征与右图那个特定位置点的所有切片中的特征值作比较，以找到特征值最接近的切片，这个切片序号回归期望反映了这个像素点在左右两图中的不一致性。

将立体估计子网络中的中间节点引出，同时取

表 1 网络节点参数

Tab. 1 Network node parameter

节点	参数	输出维度
Left image/ Right image		$3\times H\times W$
CNN_0	$3\times3,32$, stride = 2 $3\times3,32$ $3\times3,64$ Averagepool, [3,3], stride = 2	$64\times\frac{1}{4}H\times\frac{1}{4}W$
Residual $Block_{1_0}$	$1\times1,128$	$128\times\frac{1}{4}H\times\frac{1}{4}W$
Residual $Block_{1_1}$	Input = CNN_0 $1\times1,32$ $3\times3,32$ $1\times1,128$ Add Residual $Block_{1_0}$	$128\times\frac{1}{4}H\times\frac{1}{4}W$
Residual $Block_1$	[$1\times1,32$ $3\times3,32$ $1\times1,128$ Add upper layer] $\times3$	$128\times\frac{1}{4}H\times\frac{1}{4}W$
Residual $Block_{2_0}$	$1\times1,256$, stride = 2	$256\times\frac{1}{8}H\times\frac{1}{8}W$
Residual $Block_{2_1}$	Input = Residual $Block_1$ $1\times1,64$ $3\times3,64$, stride = 2, dila = 2 $1\times1,256$ Add Residual $Block_{2_0}$	$256\times\frac{1}{8}H\times\frac{1}{8}W$
Residual $Block_2$	$1\times1,64$ $3\times3,64$, dila = 2 $1\times1,256$ Add upper layer	$256\times\frac{1}{8}H\times\frac{1}{8}W$
Residual $Block_{3_0}$	$1\times1,256$	$256\times\frac{1}{8}H\times\frac{1}{8}W$
Residual $Block_{3_1}$	Input = Residual $Block_2$ $1\times1,64$ $3\times3,64$, dila = 2 $1\times1,256$ Add Residual $Block_{3_0}$	$256\times\frac{1}{8}H\times\frac{1}{8}W$
Residual $Block_3$	$1\times1,64$ $3\times3,64$, dila = 2 $1\times1,256$ Add upper layer	$256\times\frac{1}{8}H\times\frac{1}{8}W$
Residual $Block_{4_0}$	$1\times1,256$	$256\times\frac{1}{8}H\times\frac{1}{8}W$
Residual $Block_{4_1}$	Input = Residual $Block_3$ $1\times1,64$ $3\times3,64$, dila = 4 $1\times1,256$ Add Residual $Block_{4_0}$	$256\times\frac{1}{8}H\times\frac{1}{8}W$
Residual $Block_4$	$1\times1,64$ $3\times3,64$, dila = 4 $1\times1,256$ Add upper layer	$256\times\frac{1}{8}H\times\frac{1}{8}W$
Residual	Bilinear interpolation	$256\times\frac{1}{4}H\times\frac{1}{4}W$

续表 1

节点	参数	输出维度
$Branch_1$	32×32avg. pool, stride = 32 $3\times3,64$ Bilinear interpolation	$64\times\frac{1}{4}H\times\frac{1}{4}W$
$Branch_2$	16×16avg. pool, stride = 16 $3\times3,64$ Bilinear interpolation	$64\times\frac{1}{4}H\times\frac{1}{4}W$
$Branch_3$	8×8avg. pool, stride = 8 $3\times3,64$ Bilinear interpolation	$64\times\frac{1}{4}H\times\frac{1}{4}W$
$Branch_4$	4×4avg. pool, stride = 4 $3\times3,64$ Bilinear interpolation	$64\times\frac{1}{4}H\times\frac{1}{4}W$
$Conv_{512}$	Concat[Residual, $Branch_1$, $Branch_2$, $Branch_3$, $Branch_4$]	$512\times\frac{1}{4}H\times\frac{1}{4}W$
Conv	$3\times3,128$, $1\times1,32$	$32\times\frac{1}{4}H\times\frac{1}{4}W$
Cost_volume	$D=128$	$64\times\frac{1}{4}D\times\frac{1}{4}H\times\frac{1}{4}W$
3D $Conv_1$	[$3\times3\times3,32$] $\times2$	$32\times\frac{1}{4}D\times\frac{1}{4}H\times\frac{1}{4}W$
3D $Conv_2$	[$3\times3\times3,32$] $\times2$	$32\times\frac{1}{4}D\times\frac{1}{4}H\times\frac{1}{4}W$
3D $Conv_{combine}$	3D $Conv_1$ + 3D $Conv_2$	$32\times\frac{1}{4}D\times\frac{1}{4}H\times\frac{1}{4}W$
3D stack	$3\times3\times3,64$, stride = 2 $3\times3\times3,64$ $3\times3\times3,64$, stride = 2 $3\times3\times3,64$ Deconv, $3\times3\times3,64$, stride = 2 Deconv, $3\times3\times3,32$, stride = 2 Add 3D $Conv_{combine}$	$32\times\frac{1}{4}D\times\frac{1}{4}H\times\frac{1}{8}W$
$Output_{raw}$	$3\times3\times3,32$ $3\times3\times3,1$	$1\times\frac{1}{4}D\times\frac{1}{4}H\times\frac{1}{4}W$
up sample	Squeeze Bilinear interpolation	$\frac{1}{4}D\times H\times W$
regression		$H\times W$
CNN_1	Input = $Conv_{512}$ $1\times1,128$ $3\times3,128$ $1\times1,512$	$512\times\frac{1}{4}H\times\frac{1}{4}W$
CNN_2	Input = Seg_{subout} $1\times1,128$ $3\times3,128$ $1\times1,512$	$512\times\frac{1}{4}H\times\frac{1}{4}W$
fusion	Inner product[CNN_1, CNN_2]	$512\times\frac{1}{4}H\times\frac{1}{4}W$
CNN_3	$1\times1,512$ $1\times1,c_{num}$	$c_{num}\times\frac{1}{4}H\times\frac{1}{4}W$
$Seg_{upsample}$	Bilinear interpolation	$c_{num}\times H\times W$
$Seg_{softmax}$	Softmax	$c_{num}\times H\times W$
Seg_{output}	Argmax	$H\times W$
3D cloud point	Input = Seg_{output}, regression Transform	$3\times H\times W$

左图的特征提取层中间节点，传入分割子网络。两者经过融合、卷积、上采样、Softmax 和 Argmax 操作，输出像素级分类结果。

1.2　特征提取层

普通的多层残差网络对立体估计任务和图像分割任务没有很好的通用性，训练好的网络会出现立体估计效果好但分割效果不理想的结果。对梯度隔离，轮流交替训练时[15]，可以明显发现两个子网络的收敛方向不一致。用膨胀卷积[16]残差块替换 PSMNet 的普通卷积残差块，可以明显加速模型收敛，并且对于局部区域相似的特征有更好的识别能力。并且在减少层数后，性能表现并没有过多降低。

SSN 特征提取层首先使用了 3 个卷积层和 1 个池化层进行特征初步提取。再传入 4 个残差网络块融合深层与浅层特征信息。残差网络中前两个使用普通卷积，第 3 个使用膨胀数为 2 的膨胀卷积，第 4 个使用膨胀数为 4 的膨胀卷积。再使用空间金字塔池化[17]，最终输出结果计为 Conv，其尺寸大小为 $B\times32\times\frac{1}{4}H\times\frac{1}{4}W$，其中 B 为批数，H、W 为输入图像高度和宽度。立体估计与分割子网络共享特征提取层权值。

1.3　三维卷积层

PSMNet 原结构中堆叠多个沙漏结构，并且不断合并上一层的中间输出。沙漏结构是一个卷积再反卷积的过程，这样的结构可以提升模型的识别准确性，多个沙漏结构堆叠对准确性提高并不明显，但运算时间成倍增加。SSN 采用轻量级分割网络，为与之在运算速度和精度上进行匹配，对立体估计子网络进行了精简，仅使用一个沙漏结构，降低约 1% 精度，提升 2 ~ 3 倍的运行速度。

1.4　融合层

将立体估计子网络中，上采样操作之前的节点 $\mathrm{Seg}_{\mathrm{subout}}$ 引出，$\mathrm{Seg}_{\mathrm{subout}}$ 尺寸为 $B\times\frac{1}{4}D\times\frac{1}{4}H\times\frac{1}{4}W$（$D$ 为不一致性分辨率），与 Conv 的尺寸除特征维度外是一致的。同时取左图的 Conv 上一级未经卷积降维的输出节点，它的尺寸为 $B\times512\times\frac{1}{4}H\times\frac{1}{4}W$，记作 Conv_{512}。使用卷积核大小 3 × 3、深度 128 对 $\mathrm{Seg}_{\mathrm{subout}}$ 做卷积运算，得到不一致性特征图的梯度信息，再使用 1 × 1、深度 512 的卷积核做卷积运算。Conv_{512} 同样使用卷积核大小3 × 3、深度 128 的卷积核做卷积运算，再使用 1 × 1、深度 512 的卷积核做卷积运算。两者在统一尺寸后，用做内积的方式进行融合。与在特征维度堆叠的方式相比，这种方式不会扩增维度，运行速度较快。并且使得不一致性特征图对像素分割影响得更剧烈。

1.5　点云映射层

分割子网络输出像素点所对应的分类结果，记为 c，立体估计子网络输出像素点不一致图，记为 $\boldsymbol{d}$。在内存中存放两个像素坐标矩阵 $\boldsymbol{X}$、$\boldsymbol{Y}$。利用小孔成像模型转换成世界坐标系下的点云坐标。点云坐标用（$\boldsymbol{X}_r$，$\boldsymbol{Y}_r$，$\boldsymbol{Z}_r$）表示。感兴趣的物体分类号记为 c_s，比如需要筛选出路面点云，$c_s=7$；需要筛选出车辆点云，$c_s=26$。$\boldsymbol{M}$ 为筛选掩码。运算中除常值外，所有矩阵大小均为 $H\times W$，乘法运算为内积方式，在 GPU 上运行速度很快。本文中出现的矩阵与常值做比较或运算，为矩阵中所有元素与常值做比较或运算，做比较时，满足条件为“1”，否则为“0”，输出结果尺寸与原矩阵尺寸一致。

$$\boldsymbol{X}=\begin{bmatrix}0&1&\cdots&W-1\\0&1&\cdots&W-1\\\vdots&\vdots&\vdots&\vdots\\0&1&\cdots&W-1\end{bmatrix}\tag{1}$$

$$\boldsymbol{Y}=\begin{bmatrix}0&0&\cdots&0\\1&1&\cdots&1\\\vdots&\vdots&\vdots&\vdots\\H-1&H-1&\cdots&H-1\end{bmatrix}\tag{2}$$

$$\boldsymbol{M}=c_s\boldsymbol{d}\quad(\boldsymbol{d}\geqslant15)\tag{3}$$

$$\boldsymbol{D}=\frac{f_xB}{\boldsymbol{d}}\boldsymbol{M}\tag{4}$$

$$\boldsymbol{Z}_r=\frac{\boldsymbol{D}}{1+\sqrt{\left(\frac{\boldsymbol{X}-c_x}{f_x}\right)^2+\left(\frac{\boldsymbol{Y}-c_y}{f_y}\right)^2}}\tag{5}$$

$$\boldsymbol{X}_r=\frac{\boldsymbol{Z}_r(\boldsymbol{X}-c_x)}{f_x}\tag{6}$$

$$\boldsymbol{Y}_r=-\frac{\boldsymbol{Z}_r(\boldsymbol{Y}-c_y)}{f_y}\tag{7}$$

式中　f_x、f_y——相机焦距

　　c_x、c_y——相机光轴中心位置

2　模型训练及超参设置

2.1　KITTI 数据集

本文使用了 KITTI 数据集[18]中立体估计（Stereo evaluation 2015）和语义分割估计（Semantic segmentation evaluation）两类数据。立体估计数据集中包含了左右摄像头拍摄的原始图像，以及基于左图的像素级左右图不一致性表达。语义分割估计数据集中包含与立体估计数据集完全一致的左摄像头原始图像，以及左摄像头图像上的像素级语义分割

结果。无人驾驶场景关注的车辆分类标签为 26，路面分类标签为 7。

2.2 损失函数

立体估计子网络使用 $\text{smooth}_{\text{L1}}$[19] 损失函数

$$\text{loss}_{\text{stero}}(\boldsymbol{d},\hat{\boldsymbol{d}}) = \frac{1}{N}\sum_{i=1}^{N}\text{smooth}_{\text{L1}}(\boldsymbol{d}_i - \hat{\boldsymbol{d}}_i) \tag{8}$$

其中

$$\text{smooth}_{\text{L1}}(x) = \begin{cases} 0.5x^2 & (|x|<1) \\ |x| - 0.5 & (|x|\geqslant 1) \end{cases} \tag{9}$$

式中 $\hat{\boldsymbol{d}}$——深度估计标签值

$\boldsymbol{d}$——深度估计预测值

N——像素总数

分割子网络使用均方误差(Mean squared error，MSE)作为损失函数

$$\text{loss}_{\text{seg}}(\boldsymbol{c},\hat{\boldsymbol{c}}) = \frac{1}{N}\sum_{N}\left[\frac{1}{N_c}\sum_{N_c}(\boldsymbol{c}-\hat{\boldsymbol{c}})^2\right] \tag{10}$$

式中 $\hat{\boldsymbol{c}}$——分类标签值 $\boldsymbol{c}$——分类预测值

N_c——分类总数

采用联合训练的方式，构建综合损失函数 $\text{loss}_{\text{total}}$，它们的权重取值按照两个损失函数的最终收敛值的数量级以及所设计的性能偏好来选取，即

$$\text{loss}_{\text{total}} = \text{loss}_{\text{stero_match}} + 100\text{loss}_{\text{seg}} \tag{11}$$

2.3 批次正则化

子网络信息融合时，由于使用了不同类别的数据集，会出现节点输出量级上的不统一。在融合层，会导致小数信息在大数信息中湮没。在层与层之间加入批正则化(Batch normalization，BN)[20] 层，所有节点输出将归一化到 0 附近的分布，可有效解决这个问题。BN 技术被广泛用于深度学习各个领域，它可以使得梯度容易传递，加速训练速度，也最大程度地发挥所有层的价值。对于 SNN 来说，BN 是必需的。也正因为 BN 的存在，训练时，每批次训练样本应至少大于等于 3，取得越大越好，但对于显存容量要求就会越高。训练时要注意及时删除中间变量，并需要周期性地清理显存，释放显存空间。本文每批次训练样本取 4。

2.4 学习率

由于采用了 BN 技术，基础学习率 l_{base} 可以适当高一些，训练过程设置变学习率，使学习率随着迭代次数 e 增加动态降低，以保证模型更好地收敛。对于属于分割子网络以及立体估计子网络的模型参数，即特征提取网络部分，按照两倍基础学习率进行训练，使网络各环节参数收敛速度保持一致。

$$l = \begin{cases} 2l_{\text{base}} & (\text{parameter}\in \text{seg net}\cap\text{stereo net}) \\ l_{\text{base}} & (\text{其他}) \end{cases} \tag{12}$$

其中

$$l_{\text{base}} = \begin{cases} 0.001 & (e<200) \\ 0.001\left(1-\dfrac{e-200}{E-200}\right) & (200\leqslant e\leqslant 500) \end{cases} \tag{13}$$

式中 E——循环总数

2.5 收敛性判定

每次循环全样本参与训练，训练循环总数为 500。训练过程分割损失函数、立体估计损失函数、总损失函数变化见图 2。它们变化趋势一致，并随着迭代进行不断收敛最终稳定。

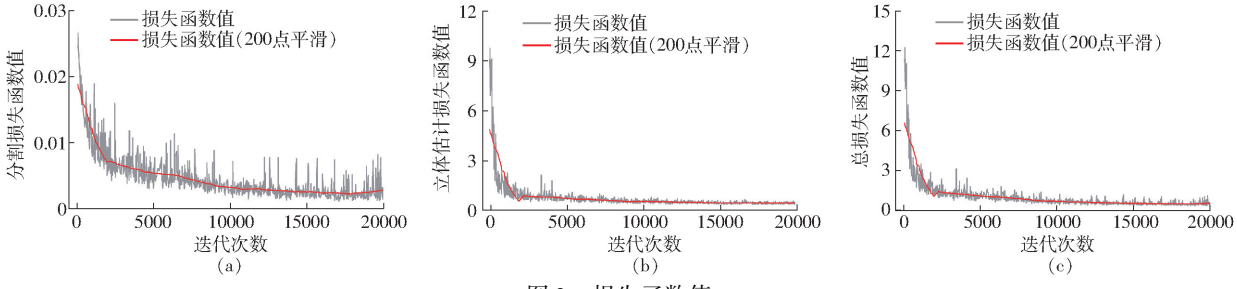

图 2 损失函数值

Fig. 2 Record of loss function

3 模型性能评价

每个全数据训练周期结束，均使用验证集测试模型准确度，以分割错误率和立体估计错误率为评价指标。

3.1 准确度

分割错误率 e_{seg}、立体估计错误率 e_{stero} 计算式为

$$e_{\text{seg}} = \frac{\sum(\boldsymbol{c}!=\hat{\boldsymbol{c}})}{\sum(\boldsymbol{c}!=\text{None})}\times 100\% \tag{14}$$

$$e_{\text{stero}} = \frac{\sum\left(\boldsymbol{d}\geqslant 15\,\&\,\dfrac{\boldsymbol{d}-\hat{\boldsymbol{d}}}{\hat{\boldsymbol{d}}}\geqslant 0.05\right)}{\sum(\boldsymbol{d}\geqslant 15)}\times 100\% \tag{15}$$

分割错误率 e_{seg}、立体估计错误率 e_{stero} 随训练循环次数变化曲线见图 3。以 $0.2e_{\text{seg}}+e_{\text{stero}}$ 最小为指标保存最佳模型。最佳模型分割错误率为 17.5%，立体估计错误率 4.5%，即分割准确率 82.5%，立体估计准确率 95.5%。

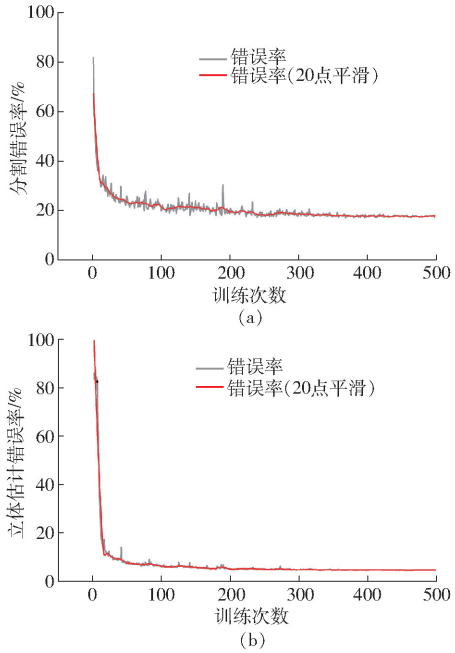

图 3　测试结果

Fig. 3　Test result

3.2　运行速度

在 1080TiGPU 设备上测试。在 1 次预运行模型后，第 2 ~4 次运算时间会极快。这可能与清除显存产生额外耗时有关。在连续运行时，保持显存容量无限是不可能的，因此对这些极快的运行次不做讨论。使模型预运行 10 次，运算速度达到稳定后，运算 100 次，记录网络从输入到主要节点的平均运算时间，具体结果见表 2。结果表明用双目相机实现点云输出可以达到 0. 135 s/帧，即每 135 ms 生成约 4. 8 万个点云坐标，其强实时性使其有很大的实际应用价值。

表 2　节点耗时

Tab. 2　Time-consuming of nodes　　ms

节点	稳定耗时	最大耗时	极快耗时
CNN/left	21	21	21
CNN/right	41	41	41
Cost_volume	49	49	49
3D $Conv_{combine}$	74	78	
3D $stack1_{1_1}$	98	110	
output1	114	119	61
regression	127	131	63
CNN_3	134	140	64
3D cloud point	135	142	

3.3　可视化效果

取验证集中一组双目图像进行可视化展示，如图 4 所示。

输入图像分辨率为 384 像素 ×1 240 像素，模型最终的输出结果为 479 232 个带 anchor 的点云坐标，anchor 存储点云所属物体类别属性，比如车或是行人。可以容易地根据 anchor 来筛选所需要的点云。对于具体应用场景，根据需求生成筛选后的点



(a) 输入左目图像

(b) 输入右目图像

(c) 分割结果可视化图

(d) 立体估计可视化图

图 4　分割和立体估计可视化

Fig. 4　Visualizations of segmentation and stereo estimation

云，可以排除非感兴趣点云的干扰，对后续控制算法实现提供巨大便利。图 5a 为筛选的路面点云，图 5b 为筛选的车辆点云，场景重现效果与图像一致。

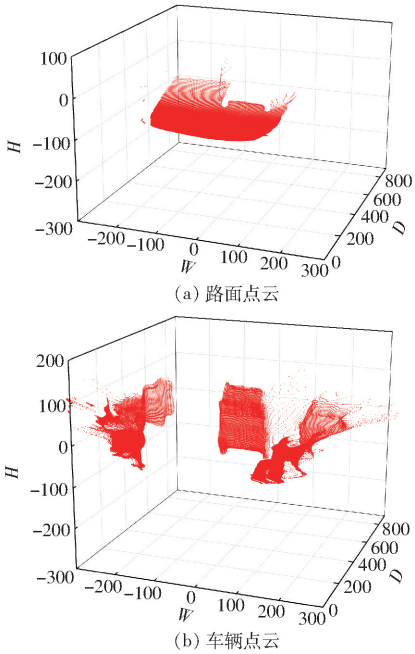

图 5　输出点云可视化

Fig. 5　Visualizations of cloud points

4　结论

(1)提出了一种同时实现双目立体估计与图像

分割的联合训练网络,即空间语义网络。

(2)利用深度学习技术,通过数据训练模型,实现了仅依靠双目相机实现输出语义点云的功能。

(3)设计的模型经过训练可以达到分割准确率 82.5%;针对近点,以立体估计误差 5% 以内为准确,立体估计准确率可以达到 95.5%。

(4)设计的模型运行速度可以达到 0.135 s/帧,即每 135 ms 生成约 4.8 万个点云坐标,接近低速工况下实时处理要求。

参 考 文 献

[1] 陆德彪,郭子明,蔡伯根,等. 基于深度数据的车辆目标检测与跟踪方法[J]. 交通运输系统工程与信息,2018, 18(3):59 - 66.
LU Debiao, GUO Ziming, CAI Bogen, et al. A vehicle detection and tracking method based on range data [J]. Journal of Transportation Systems Engineering and Information Technology, 2018, 18(3): 59 - 66. (in Chinese)

[2] 王新竹,李骏,李红建,等. 基于三维激光雷达和深度图像的自动驾驶汽车障碍物检测方法[J]. 吉林大学学报(工学版), 2016,46(2): 360 - 365.
WANG Xinzhu, LI Jun, LI Hongjian, et al. Obstacle detection based on 3D laser scanner and range image for intelligent vehicle [J]. Journal of Jilin University (Engineering and Technology Edition), 2016,46(2): 360 - 365. (in Chinese)

[3] 程曼,蔡振江,WANG Ning,等. 基于地面激光雷达的田间花生冠层高度测量系统研制[J]. 农业工程学报, 2019,35(1): 180 - 187.
CHENG Man, CAI Zhenjiang, WANG Ning, et al. System design for peanut canopy height information acquisition based on LiDAR[J]. Transactions of the CSAE, 2019, 35(1): 180 - 187. (in Chinese)

[4] 张漫,苗艳龙,仇瑞承,等. 基于车载三维激光雷达的玉米叶面积指数测量[J]. 农业机械学报, 2019,50(6):12 - 21.
ZHANG Man, MIAO Yanlong, QIU Ruicheng, et al. Maize leaf area index measurement based on vehicle 3D LiDAR [J]. Transactions of the Chinese Society for Agricultural Machinery,2019,50(6):12 - 21. (in Chinese)

[5] 张雪,郭彩玲,宗泽,等. 基于轮廓分割的草莓叶片三维建模[J]. 农业工程学报, 2017,33(增刊 1): 206 - 211.
ZHANG Xue, GUO Cailing, ZONG Ze, et al. 3D reconstruction of strawberry leaves based on contour segmentation [J]. Transactions of the CSAE, 2017, 33(Supp. 1): 206 - 211. (in Chinese)

[6] 徐胜勇,卢昆,潘礼礼,等. 基于 RGB - D 相机的油菜分枝三维重构与角果识别定位[J]. 农业机械学报, 2019,50(2): 21 - 27.
XU Shengyong, LU Kun, PAN Lili, et al. 3D reconstruction of rape branch and pod recognition based on RGB - D camera [J]. Transactions of the Chinese Society for Agricultural Machinery, 2019,50(2): 21 - 27. (in Chinese)

[7] 劳彩莲,杨瀚,李鹏,等. 基于消费级深度相机的玉米植株三维重建[J/OL]. 农业机械学报. http://kns.cnki.net/kcms/detail/11.1964.S.20190412.1111.004.html.
LAO Cailian, YANG Han, LI Peng, et al. 3D reconstruction of maize plants based on consumer depth camera [J/OL]. Transactions of the Chinese Society for Agricultural Machinery. http://kns.cnki.net/kcms/detail/11.1964.S.20190412.1111.004.html. (in Chinese)

[8] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, 2015, 521(7553):436.

[9] LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 39(4):640 - 651.

[10] ZEILER M D, TAYLOR G W, FERGUS R. Adaptive deconvolutional networks for mid and high level feature learning[C]// 2011 International Conference on Computer Vision. IEEE, 2011.

[11] ZHAO H, QI X, SHEN X, et al. ICNet for real-time semantic segmentation on high-resolution images[C]//Lecture Notes in Computer Science, 2017.

[12] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition [C]//2016 IEEE Conference on Computer Vision and Pattern Recognition,2016.

[13] CHANG J R, CHEN Y S. Pyramid stereo matching network [C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Computer Vision and Pattern Recognition (CVPR), 2018.

[14] CHEN L, YANG Z, MA J, et al. Driving scene perception network: real-time joint detection, depth estimation and semantic segmentation[C]//2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE Computer Society, 2018.

[15] RADFORD A, METZ L, CHINTALA S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv:1511.06434, 2015.

[16] CHEN L C, PAPANDREOU G, KOKKINOS I, et al. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, 40(4):834 - 848.

[17] HE K, ZHANG X, REN S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 37(9):1904 - 1916.

[18] ABU ALHAIJA H, MUSTIKOVEL S K, MESCHEDER L, et al. Augmented reality meets computer vision: efficient data generation for urban driving scenes[J]. International Journal of Computer Vision, 2018, 126(9):961 - 972.

[19] GIRSHICK R. Fast R - CNN[C]//2015 IEEE International Conference on Computer Vision (ICCV). IEEE, 2015.

[20] LOFFE S, SZEGEDY C. Batch normalization: accelerating deep network training by reducing internal covariate shift[C]// International Conference on International Conference on Machine Learning. JMLR, 2015.