



(12)发明专利申请

(10)申请公布号 CN 110531340 A

(43)申请公布日 2019.12.03

(21)申请号 201910776978.X

(22)申请日 2019.08.22

(71)申请人 吴文吉

地址 362211 福建省泉州市晋江市东石镇  
坑园村东区10号

申请人 叶华

(72)发明人 吴文吉 叶华

(74)专利代理机构 河南大象律师事务所 41129

代理人 尹周

(51) Int. Cl.

G01S 7/48(2006.01)

G01S 17/42(2006.01)

G01S 17/50(2006.01)

G06N 3/04(2006.01)

G06N 3/08(2006.01)

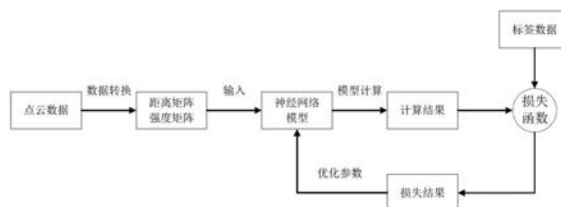
权利要求书1页 说明书6页 附图1页

(54)发明名称

一种激光雷达点云数据的基于深度学习的识别处理方法

(57)摘要

本发明属于一种激光雷达点云数据的基于深度学习的识别处理方法,使用训练好的神经网络模型进行识别时,以yolov2的计算方法对神经网络模型的结果进行计算,在计算结果的宽度和高度时,分别要用到一个该类别对应的锚定值,锚定值优选标注数据中的所有同一类别物体的平均宽度和高度,本发明避免了点云数据的预处理过程,也能识别出物体类别,通过把点云数据转换为距离矩阵和反射光强度矩阵形式作为深度学习神经网络的输入,保持了点云数据的三维特征,从而保证小物体的较高识别率,并且把误识别率控制在很低的水平,最终得到的结果是矩阵维度下的结果,可以根据实际场景进行转换。



1. 一种激光雷达点云数据的基于深度学习的识别处理方法,其特征在於:包括以下步骤:

步骤1)、获取原始点云数据和标注数据,原始点云数据包括有每个点的角度、距离和反射光强度值,原始点云数据以帧为单位,一帧数据为激光雷达转一圈扫过 $360^{\circ}$ 所产生的点云数据;

步骤2)、将原始点云数据的距离和反射光强度值,以 $360$ 角度为水平方向,激光雷达的线数为垂直方向分别排成一个距离矩阵和一个反射光强度矩阵;

步骤3)、修正距离矩阵和反射光强度矩阵的维度;将距离矩阵和反射光强度矩阵的维度修正为设定的固定维度;

步骤4)、强化距离矩阵和反射光强度矩阵中处于边缘对象的识别;

步骤5)、以yolov1或yolov2的神经网络模型搭建卷积神经网络;

步骤6)、以yolov1的计算方法计算损失函数;进行神经网络模型训练;

步骤7)、损失函数结果稳定后,在各卷积层使用L2正则化,再次对神经网络进行训练;

步骤8)、损失函数结果再次稳定后,即可使用训练好的神经网络模型进行识别。

2. 根据权利要求1所述的激光雷达点云数据的基于深度学习的识别处理方法,其特征在於:在步骤3)中,修正距离矩阵或反射光强度矩阵的维度小于固定维度时,向修正距离矩阵或反射光强度矩阵两侧填充进数据0;修正距离矩阵或反射光强度矩阵的维度大于固定维度时,在数据中,随机删去大于固定维度数值的维度,并将所删除维度的左右两侧的数据修改为该数据与删除维度数据的平均值。

3. 根据权利要求1所述的激光雷达点云数据的基于深度学习的识别处理方法,其特征在於:在步骤4)中,向距离矩阵或反射光强度矩阵的靠近地面的边缘处填充2行数据0,增大距离矩阵或反射光强度矩阵,使物体的点云数据在距离矩阵或反射光强度矩阵中往中间移动两行,强化处于边缘的物体识别。

4. 根据权利要求1所述的激光雷达点云数据的基于深度学习的识别处理方法,其特征在於:在步骤5)中,在搭建神经网络模型的卷积处理中,通过池化层缩小矩阵长的维度,通过卷积缩小矩阵宽的维度,使强化距离矩阵或反射光强度矩阵的最终结果维度变为指定值;删除全连接层,使用卷积得出的结果直接进行预测。

5. 根据权利要求1所述的激光雷达点云数据的基于深度学习的识别处理方法,其特征在於:在步骤6)中神经网络模型训练过程中使用Adam优化算法进行加速收敛。

6. 根据权利要求1所述的激光雷达点云数据的基于深度学习的识别处理方法,其特征在於:在步骤6)神经网络模型训练过程中,包括有N层卷积,其中 $1\sim N-1$ 层卷积使用Batch Normalization,防止神经网络模型参数过拟合。

7. 根据权利要求1所述的激光雷达点云数据的基于深度学习的识别处理方法,其特征在於:使用训练好的神经网络模型进行识别时,以yolov2的计算方法对神经网络模型的结果进行计算,在计算结果的宽度和高度时,分别要用到一个该类别对应的锚定值,锚定值优选标注数据中的所有同一类别物体的平均宽度和高度。

## 一种激光雷达点云数据的基于深度学习的识别处理方法

### 技术领域

[0001] 本发明涉及深度学习领域,尤其涉及一种激光雷达点云数据的基于深度学习的识别处理方法。

### 背景技术

[0002] LiDAR——Light Detection And Ranging,即激光探测与测量,也就是激光雷达,是以发射激光束探测目标的位置、速度等特征量的雷达系统。其工作原理是向目标发射探测信号(激光束),然后将接收到的从目标反射回来的信号(目标回波)与发射信号进行比较,作适当处理后,就可获得目标的有关信息,如目标距离、方位、高度、速度、姿态、甚至形状等参数,其所测得的数据为数字表面模型(Digital Surface Model, DSM)的离散点表示,数据中含有空间三维信息和激光强度信息。根据激光束的多少,激光雷达可以分为单线和多线,其中多线的包括16线、20线、32线、64线,甚至128线等种类。激光雷达在无人驾驶、车路协同、机器人导航等领域,作为关键传感器使用。

[0003] 激光雷达产生的数据是离散的点云数据,在无人驾驶和车路协同等领域,需要将点云数据运用一些算法进行处理,识别出其中的汽车、行人等对象,并用算法计算出这些对象的位置。点云数据处理算法,通常分为有监督算法,如基于深度学习的算法,还有无监督算法,如基于聚类的算法。随着人工智能技术和理论的发展,基于深度学习的算法,越来越受到研究人员的重视,其效果也越来越好。无监督的算法,实施过程中通常需要进行过程繁琐的预处理,比如先拟合地面,然后过滤地面的点,再把道路周围与驾驶无关的点过滤掉,然后才能进行识别,预处理的效果对最终识别的效果会有很大影响,并且无监督的算法不能直接识别出对象类别。而现有的深度学习算法,在把点云数据转换为深度学习神经网络的输入方面,没有很好的方法,导致对于小的物体识别率很低,基本识别不出来,也存在较高的误识别率,会把背景点云识别为目标点。

### 发明内容

[0004] 本发明目的在于克服现有识别算法中存在的不足而提供一种激光雷达点云数据的基于深度学习的识别处理方法。

[0005] 本发明的目的是这样实现的:一种激光雷达点云数据的基于深度学习的识别处理方法,包括以下步骤:

步骤1)、获取原始点云数据和标注数据,原始点云数据包括有每个点的角度、距离和反射光强度值,原始点云数据以帧为单位,一帧数据为激光雷达转一圈扫过 $360^{\circ}$ 所产生的点云数据;

步骤2)、将原始点云数据的距离和反射光强度值,以 $360$ 角度为水平方向,激光雷达的线数为垂直方向分别排成一个距离矩阵和一个反射光强度矩阵;

步骤3)、修正距离矩阵和反射光强度矩阵的维度;将距离矩阵和反射光强度矩阵的维度修正为设定的固定维度;

- 步骤4)、强化距离矩阵和反射光强度矩阵中处于边缘对象的识别;
- 步骤5)、以yolov1或yolov2的神经网络模型搭建卷积神经网络;
- 步骤6)、以yolov1的计算方法计算损失函数;进行神经网络模型训练;
- 步骤7)、损失函数结果稳定后,在各卷积层使用L2正则化,再次对神经网络进行训练;
- 步骤8)、损失函数结果再次稳定后,即可使用训练好的神经网络模型进行识别。

[0006] 在步骤3)中,修正距离矩阵或反射光强度矩阵的维度小于固定维度时,向修正距离矩阵或反射光强度矩阵两侧填充进数据0;修正距离矩阵或反射光强度矩阵的维度大于固定维度时,在数据中,随机删去大于固定维度数值的维度,并将所删除维度的左右两侧的数据修改为该数据与删除维度数据的平均值。

[0007] 在步骤4)中,向距离矩阵或反射光强度矩阵的靠近地面的边缘处填充2行数据0,增大距离矩阵或反射光强度矩阵,使物体的点云数据在距离矩阵或反射光强度矩阵中往中间移动两行,强化处于边缘的物体识别。

[0008] 在步骤5)中,在搭建神经网络模型的卷积处理中,通过池化层缩小矩阵长的维度,通过卷积缩小矩阵宽的维度,使强化距离矩阵或反射光强度矩阵的最终结果维度变为指定值;删除全连接层,使用卷积得出的结果直接进行预测。

[0009] 在步骤6)神经网络模型训练过程中使用Adam优化算法进行加速收敛。

[0010] 在步骤6)神经网络模型训练过程中,包括有N层卷积,其中1~N-1层卷积使用Batch Normalization,防止神经网络模型参数过拟合。

[0011] 使用训练好的神经网络模型进行识别时,以yolov2的计算方法对神经网络模型的结果进行计算,在计算结果的宽度和高度时,分别要用到一个该类别对应的锚定值,锚定值优选标注数据中的所有同一类别物体的平均宽度和高度。

[0012] 本发明的优点好处:

本发明避免了点云数据的预处理过程,也能识别出物体类别,通过把点云数据转换为距离矩阵和反射光强度矩阵形式作为深度学习神经网络的输入,保持了点云数据的三维特征,从而保证小物体的较高识别率,并且把误识别率控制在很低的水平。

## 附图说明

[0013] 图1为本发明神经网络模型训练的逻辑框图。

## 具体实施方式

[0014] 实施例1:一种激光雷达点云数据的基于深度学习的识别处理方法,包括以下步骤:

步骤1)、获取原始点云数据和标注数据,原始点云数据包括每个点的角度、距离和反射光强度值,原始点云数据以帧为单位,一帧数据为激光雷达转一圈扫过360°所产生的点云数据;

如下所示为16线激光雷达原始点云数据的部分实例,以第一个数据为例,333.03表示水平旋转角度,后面的460和9分别表示该水平角度下-15°所在的激光线扫描得到的距离和发射光强度,430和54则是-13°激光线对应的距离和反射光强度,后面的数据对应的激光线角度依次为-11°, -9°, -7°, -5°, -3°, -1°, 1°, 3°, 5°, 7°, 9°, 11°, 13°, 15°。总结一行原始点

云数据的格式为:1个角度+16组距离和反射光强度。

[0015] 333.03 460 9 430 54 470 9 410 10 490 9 0 0 400 12 0 0 430 21 0 0 440  
19 0 0 390 25 0 0 440 47 0 0

333.20 460 9 410 43 470 10 430 10 450 8 0 0 420 10 0 0 430 18 0 0 410 16  
0 0 430 24 0 0 430 47 0 0

333.38 440 10 420 67 450 11 370 12 490 9 0 0 430 14 0 0 440 24 0 0 430 23  
0 0 400 34 0 0 440 65 0 0

步骤2)、将原始点云数据的距离和反射光强度值,以360角度为水平方向,激光雷达的线数为垂直方向分别排成一个距离矩阵和一个反射光强度矩阵;以16线激光雷达为例,假如一帧点云数据在360圆周方向有2010个角度,则矩阵形状为(16,2010),把每个点云的距离和反射光强度数据,按照点云的角度和所在的线数,填入矩阵,这样就能得到2个(16,2010)的矩阵,一个是距离矩阵,另一个是反射光强度矩阵。下面给出距离矩阵和反射光强度矩阵的例子,限于篇幅,例子中数据只是原矩阵一部分;

$$\text{距离矩阵 } D = \begin{bmatrix} 430 & 440 & 530 & 430 & 470 & \dots \\ 430 & 380 & 380 & 430 & 390 & \dots \\ 470 & 490 & 460 & 480 & 440 & \dots \\ 0 & 0 & 430 & 330 & 430 & \dots \\ & & & \dots & & \dots \end{bmatrix},$$

$$\text{强度矩阵 } I = \begin{bmatrix} 5 & 6 & 6 & 7 & 7 & \dots \\ 5 & 7 & 6 & 13 & 9 & \dots \\ 6 & 7 & 6 & 8 & 7 & \dots \\ 0 & 0 & 5 & 5 & 6 & \dots \\ & & & \dots & & \dots \end{bmatrix}.$$

[0016] 步骤3)、修正距离矩阵和反射光强度矩阵的维度,将距离矩阵和反射光强度矩阵的维度修正为设定的固定维度,这个固定维度是预先进行设定的,在本申请实施例中,可以预先将固定维度设定为2010。与角度个数是对应的。这里我们设定的固定维度是2010。一帧点云数据不是固定的,会出现比2010少或者多的情况,但是深度学习需要形状固定的输入数据。如果比2010少,比如只有2007个角度,则把缺少的3列数据分别在2007两侧,用0填充,由于3是奇数,则默认右边比左边多填充,比如左边填充1列0,右边填充2列0,这样凑成2010列。如果比2010多,比如有2015个角度,由于雷达的角度数据比较密集,两个角度之间相差约 $0.18^\circ$ ,去掉一个角度的数据,对于识别效果影响不大。所以可以从2015个角度中,随机去掉5个角度的数据,这5个角度要间隔一定距离,不要靠太密。对于每个去掉的角度,假如去掉的是第100个角度,则相邻第99个角度的数据要改为原来第99个角度的数据和去掉的第100个角度的数据的平均值,同理,相邻第101个角度的数据也要改为原来第101个角度的数据和去掉的第100个角度的数据的平均值。2010这个值,可以根据雷达一帧数据的平均大小而更改,比如改为2000或者2050,优选是2010。下面是填充后矩阵的例子,原矩阵有2005个角度,在左边填充2列,右边填充3列,变成2010列,限于篇幅,数据只是一部分。

$$\text{距离矩阵 } D = \begin{bmatrix} 0 & 0 & 430 & 440 & 530 & 430 & 470 & \dots & 430 & 0 & 0 & 0 \\ 0 & 0 & 430 & 380 & 380 & 430 & 390 & \dots & 450 & 0 & 0 & 0 \\ 0 & 0 & 470 & 490 & 460 & 480 & 440 & \dots & 470 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 430 & 330 & 430 & \dots & 420 & 0 & 0 & 0 \\ \dots & & & & & & & & & & & \end{bmatrix},$$

$$\text{强度矩阵 } I = \begin{bmatrix} 0 & 0 & 5 & 6 & 6 & 7 & 7 & \dots & 5 & 0 & 0 & 0 \\ 0 & 0 & 5 & 7 & 6 & 1 & 3 & 9 & \dots & 6 & 0 & 0 & 0 \\ 0 & 0 & 6 & 7 & 6 & 8 & 7 & \dots & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 5 & 6 & \dots & 5 & 0 & 0 & 0 \\ \dots & & & & & & & & & & & \end{bmatrix}$$

[0017] 步骤4)、强化距离矩阵和反射光强度矩阵中处于边缘对象的识别。神经网络主要运算为卷积运算,而卷积运算容易忽略矩阵边缘的数据,但激光雷达在实际使用中,检测到的物体刚好主要位于下面的激光束位置,因而物体经常处于靠近矩阵边缘的位置,为了防止处于边缘的物体被忽略,在靠近物体的矩阵边缘处填充2行0,矩阵大小变为(18,2010),让物体的点云数据在矩阵中往中间移动了两行。由此便可得到固定大小的距离矩阵和强度矩阵数据,以此作为深度学习神经网络的输入。下面给出矩阵强化之后的例子,分别在上面增加了2行0,限于篇幅,数据只是一部分。

$$\text{距离矩阵 } D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 430 & 440 & 530 & 430 & 470 & \dots & 430 & 0 & 0 & 0 \\ 0 & 0 & 430 & 380 & 380 & 430 & 390 & \dots & 450 & 0 & 0 & 0 \\ 0 & 0 & 470 & 490 & 460 & 480 & 440 & \dots & 470 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 430 & 330 & 430 & \dots & 420 & 0 & 0 & 0 \\ \dots & & & & & & & & & & & \end{bmatrix},$$

$$\text{强度矩阵 } I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 6 & 6 & 7 & 7 & \dots & 5 & 0 & 0 & 0 \\ 0 & 0 & 5 & 7 & 6 & 1 & 3 & 9 & \dots & 6 & 0 & 0 & 0 \\ 0 & 0 & 6 & 7 & 6 & 8 & 7 & \dots & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 5 & 6 & \dots & 5 & 0 & 0 & 0 \\ \dots & & & & & & & & & & & \end{bmatrix}$$

[0018] 步骤5)、以yolov1或yolov2的神经网络模型搭建卷积神经网络。yolov1的卷积运算的输入是方形的矩阵,而本发明的输入矩阵形状属于长条形,因而卷积处理过程中,通过池化层在长的维度缩小尺寸,宽的维度上尺寸的缩小通过卷积本身来实现;去掉全连接层,直接用卷积的结果做预测;卷积最终输出结果大小可以根据实际使用场景调整,总体调整策略为:结果形状越大,计算量越大,但是预测结果也越准确,对于16线激光雷达点云数据,



优选的输出单元数量为 $3*57$ ，即距离矩阵和反射光强度矩阵的维度从输入的 $(18, 2010)$ ，经过神经网络处理后变为 $(3, 57)$ 。

[0019] 步骤6)、以yolov1的计算方法计算损失函数，进行神经网络模型训练，模型可以不需要预训练，直接用标注好的点云数据去训练，也能收敛。神经网络模型训练过程中使用Adam优化算法进行加速收敛。训练过程中，包括有N层卷积，其中 $1\sim N-1$ 层卷积使用Batch Normalization，防止神经网络模型参数过拟合。既除了最后一层卷积不使用Batch Normalization，其他卷积层都使用Batch Normalization，防止神经网络模型参数过拟合；Batch的大小根据计算机内存和显存大小而定，在计算机性能可以承受的情况下越大越好。优选的值是16。神经网络模型结果包含的信息数量有置信度(1个数据)、预测结果位置x和y(2个数据)、预测结果尺寸宽度w和高度h(2个数据)、预测类别。假如预测类别有3个：非机动车，汽车，行人，则每个单元的结果会有 $1+2+2+3=8$ 个数据。每个单元可以输出1个预测框，也可以输出多个预测框然后进行筛选，每个单元输出一个预测框时，最终卷积神经网络输出维度参考值为 $3*57*8$ 。标注数据的格式要与模型训练的结果对应，也应该是 $3*57*8$ ，格式转换参考yolov1的方式转换，下面分别给出一个有物体的标注，和没有物体的标注示例：

有物体的单元标注数据示例：

$$\begin{bmatrix} 1 \\ 0.52 \\ 0.63 \\ 1.2 \\ 2.1 \\ 1 \\ 0 \\ 0 \end{bmatrix},$$

无物体的单元标注数据示例：

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

[0020] 步骤7)、损失函数结果稳定后，在各卷积层使用L2正则化；再次对神经网络进行训练；训练初期，为了保证模型收敛，可以先不加入L2正则化，优选的学习率是0.001。等到损失函数结果基本稳定时，再在各卷积层加上L2正则化，优选的L2正则化系数为0.01。

[0021] 步骤8)、损失函数结果再次稳定后，即可使用训练好的神经网络模型进行识别。识别时同样把原始点云数据转换为距离矩阵和反射光强度矩阵，并输入神经网络进行计算。

[0022] 使用训练好的神经网络模型进行识别时，以yolov2的计算方法对神经网络模型的结果进行计算，在计算结果的宽度和高度时，分别要用到一个该类别对应的锚定值，这个锚定值优选是标注数据中的所有同一类别物体的平均宽度和高度，比如计算行人的高度时，

要用到的锚定值,就是标注数据中所有行人的高度的平均值;然后对所有单元的预测结果概率进行排序,取概率大于预先设定的阈值的结果,进行非极大值抑制计算,优选的阈值为0.5。最终得到的结果是矩阵维度下的结果,可以根据实际场景进行转换,比如转换为三维笛卡尔坐标系下的结果。

[0023] 上述实例仅为本发明的优选实例而已,并不用以限制本发明,对于本领域的技术人员来说,本发明可以有各种更改和变化,凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。



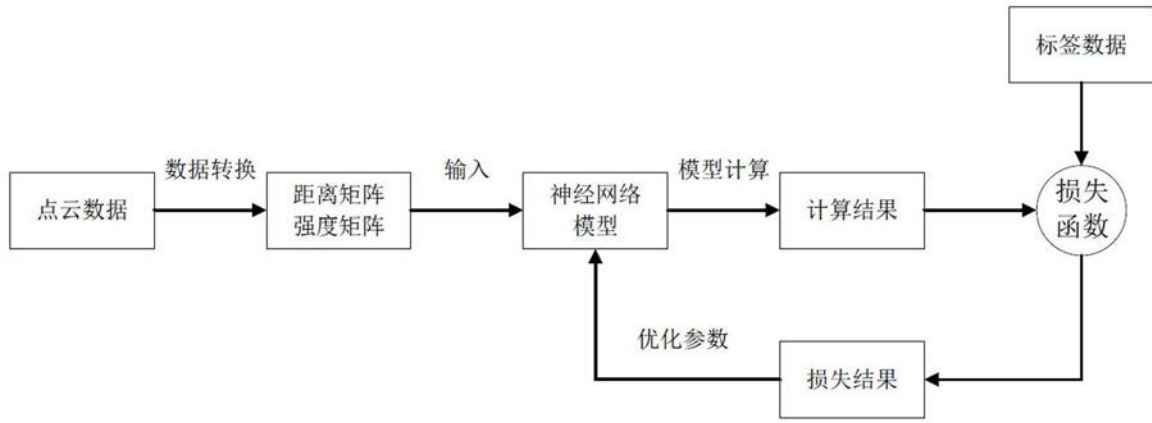


图1