



(12)发明专利申请

(10)申请公布号 CN 108629231 A

(43)申请公布日 2018. 10. 09

(21)申请号 201710157992.2

(22)申请日 2017.03.16

(71)申请人 百度在线网络技术(北京)有限公司
地址 100085 北京市海淀区上地十街10号
百度大厦

(72)发明人 陈东明 闫鹤 王昊 王亮

(74)专利代理机构 北京鸿德海业知识产权代理
事务所(普通合伙) 11412
代理人 袁媛

(51) Int. Cl.

G06K 9/00(2006.01)

G06K 9/62(2006.01)

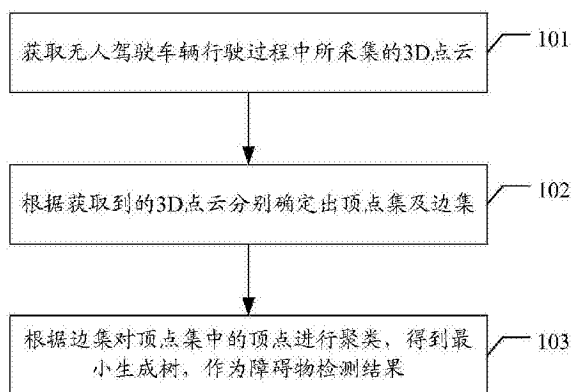
权利要求书4页 说明书9页 附图4页

(54)发明名称

障碍物检测方法、装置、设备及存储介质

(57)摘要

本发明公开了障碍物检测方法、装置、设备及存储介质,其中方法包括:获取无人驾驶车辆行驶过程中所采集的3D点云;根据3D点云分别确定出顶点集及边集;根据所述边集对顶点集中的顶点进行聚类,得到最小生成树,作为障碍物检测结果。应用本发明所述方案,能够减少工作量和提高检测结果的准确性等。



1. 一种障碍物检测方法,其特征在于,包括:
获取无人驾驶车辆行驶过程中所采集的3D点云;
根据所述3D点云分别确定出顶点集及边集;
根据所述边集对所述顶点集中的顶点进行聚类,得到最小生成树,作为障碍物检测结果。

2. 根据权利要求1所述的方法,其特征在于,
所述根据所述3D点云分别确定出顶点集及边集包括:
将所述3D点云中的每个点分别作为一个顶点,利用所有顶点组成所述顶点集;
针对每个顶点,分别确定出位于所述顶点周围预定范围内的各顶点,并分别计算出确定出的各顶点与所述顶点之间的欧式距离;

针对每个欧式距离,分别将所述欧式距离对应的两个顶点之间的连线作为一条边,并将每条边对应的欧式距离作为边的长度,利用所有边组成所述边集。

3. 根据权利要求2所述的方法,其特征在于,
所述根据所述边集对所述顶点集中的顶点进行聚类,得到最小生成树包括:
初始状态下,分别将所述顶点集中的每个顶点作为一个类;
将所述边集中的各条边按照长度从小到大的顺序进行排序,并按照排序后由先到后的顺序,依次对各条边进行以下处理:

确定所述边是否满足填加条件,如果是,则将所述边对应的两个顶点相连,进行类的聚合;

当所述边集中的各条边均处理完成后,得到聚合出的所述最小生成树。

4. 根据权利要求3所述的方法,其特征在于,
所述确定所述边是否满足填加条件包括:
分别计算出所述边对应的两个顶点所在的类的阈值;
如果所述边的长度同时小于计算出的两个阈值,且,如果将所述边对应的两个顶点相连后不会出现环,则确定所述边满足填加条件。

5. 根据权利要求4所述的方法,其特征在于,
所述分别计算出所述边对应的两个顶点所在的类的阈值包括:
针对所述边对应的每个顶点,分别进行以下处理:
确定出所述顶点所在的类中包括的顶点数Num;
确定出所述顶点所在的类中长度最大的边的长度L;
用预先设定的常数K除以Num,并将得到的商与所述L相加,将相加之和作为所述顶点所在的类的阈值。

6. 根据权利要求1所述的方法,其特征在于,
所述根据所述3D点云分别确定出顶点集及边集包括:
对所述3D点云进行下采样,根据下采样结果确定出所述顶点集及所述边集。

7. 根据权利要求6所述的方法,其特征在于,
所述对所述3D点云进行下采样,根据下采样结果确定出所述顶点集及所述边集包括:
将所述3D点云投影到3D网格上;
针对每个网格,分别计算出所述网格中的各点的均值,并将所述均值对应的点作为一

个顶点,利用所有顶点组成所述顶点集;

针对每个顶点,分别确定出位于所述顶点周围预定范围内的各顶点,并分别计算出确定出的各顶点与所述顶点之间的欧式距离;

针对每个欧式距离,分别将所述欧式距离对应的两个顶点之间的连线作为一条边,并将每条边对应的欧式距离作为边的长度,利用所有边组成所述边集。

8. 根据权利要求1所述的方法,其特征在于,

所述根据所述3D点云分别确定出顶点集及边集包括:

根据所述3D点云生成高度图,根据所述高度图确定出所述顶点集及所述边集。

9. 根据权利要求8所述的方法,其特征在于,

所述根据所述3D点云生成高度图,根据所述高度图确定出所述顶点集及所述边集包括:

将所述3D点云投影到与地平面平行的二维网格上;

针对每个网格,分别计算出所述网格对应的柱状体中的各点在x方向和y方向上的均值,并选出所述网格对应的柱状体中的各点在z方向上的最大值,将所述均值对应的点作为一个像素点,将所述最大值作为所述像素点对应的高度值,得到所述高度图;

将每个像素点分别作为一个顶点,利用所有顶点组成所述顶点集;

针对每个顶点,分别利用x方向和y方向上的取值以及所述高度值,计算出所述顶点的N*N邻域内的其他各顶点与所述顶点之间的欧式距离,N为大于1的奇数;

针对每个欧式距离,分别将所述欧式距离对应的两个顶点之间的连线作为一条边,并将每条边对应的欧式距离作为边的长度,利用所有边组成所述边集。

10. 一种障碍物检测装置,其特征在于,包括:获取单元及检测单元;

所述获取单元,用于获取无人驾驶车辆行驶过程中所采集的3D点云,并发送给所述检测单元;

所述检测单元,用于根据所述3D点云分别确定出顶点集及边集,并根据所述边集对所述顶点集中的顶点进行聚类,得到最小生成树,作为障碍物检测结果。

11. 根据权利要求10所述的装置,其特征在于,

所述检测单元中包括:第一处理子单元以及第二处理子单元;

所述第一处理子单元,用于将所述3D点云中的每个点分别作为一个顶点,利用所有顶点组成所述顶点集;

针对每个顶点,分别确定出位于所述顶点周围预定范围内的各顶点,并分别计算出确定出的各顶点与所述顶点之间的欧式距离;

针对每个欧式距离,分别将所述欧式距离对应的两个顶点之间的连线作为一条边,并将每条边对应的欧式距离作为边的长度,利用所有边组成所述边集;

所述第二处理子单元,用于根据所述边集对所述顶点集中的顶点进行聚类,得到最小生成树,作为障碍物检测结果。

12. 根据权利要求11所述的装置,其特征在于,

所述第二处理子单元具体用于:

在初始状态下,分别将所述顶点集中的每个顶点作为一个类;

将所述边集中的各条边按照长度从小到大的顺序进行排序,并按照排序后由先到后的

顺序,依次对各条边进行以下处理:

确定所述边是否满足填加条件,如果是,则将所述边对应的两个顶点相连,进行类的聚合;

当所述边集中的各条边均处理完成后,得到聚合出的所述最小生成树。

13. 根据权利要求12所述的装置,其特征在于,

所述满足填加条件包括:

所述边的长度同时小于所述边对应的两个顶点所在的类的阈值,且,将所述边对应的两个顶点相连后不会出现环。

14. 根据权利要求13所述的装置,其特征在于,

每个顶点所在的类的阈值= $L+K/Num$;

其中,Num表示所述顶点所在的类中包括的顶点数;

L表示所述顶点所在的类中长度最大的边的长度;

K表示预先设定的常数。

15. 根据权利要求11所述的装置,其特征在于,

所述第一处理子单元进一步用于,

对所述3D点云进行下采样,根据下采样结果确定出所述顶点集及所述边集。

16. 根据权利要求15所述的装置,其特征在于,

所述第一处理子单元具体用于:

将所述3D点云投影到3D网格上;

针对每个网格,分别计算出所述网格中的各点的均值,并将所述均值对应的点作为一个顶点,利用所有顶点组成所述顶点集;

针对每个顶点,分别确定出位于所述顶点周围预定范围内的各顶点,并分别计算出确定出的各顶点与所述顶点之间的欧式距离;

针对每个欧式距离,分别将所述欧式距离对应的两个顶点之间的连线作为一条边,并将每条边对应的欧式距离作为边的长度,利用所有边组成所述边集。

17. 根据权利要求11所述的装置,其特征在于,

所述第一处理子单元进一步用于,

根据所述3D点云生成高度图,根据所述高度图确定出所述顶点集及所述边集。

18. 根据权利要求17所述的装置,其特征在于,

所述第一处理子单元具体用于:

将所述3D点云投影到与地平面平行的二维网格上;

针对每个网格,分别计算出所述网格对应的柱状体中的各点在x方向和y方向上的均值,并选出所述网格对应的柱状体中的各点在z方向上的最大值,将所述均值对应的点作为一个像素点,将所述最大值作为所述像素点对应的高度值,得到所述高度图;

将每个像素点分别作为一个顶点,利用所有顶点组成所述顶点集;

针对每个顶点,分别利用x方向和y方向上的取值以及所述高度值,计算出所述顶点的 $N*N$ 邻域内的其他各顶点与所述顶点之间的欧式距离, N 为大于1的奇数;

针对每个欧式距离,分别将所述欧式距离对应的两个顶点之间的连线作为一条边,并将每条边对应的欧式距离作为边的长度,利用所有边组成所述边集。

19. 一种计算机设备,包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述程序时实现如权利要求1~9中任一项所述的方法。

20. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述程序被处理器执行时实现如权利要求1~9中任一项所述的方法。

障碍物检测方法、装置、设备及存储介质

【技术领域】

[0001] 本发明涉及无人驾驶车辆技术，特别涉及障碍物检测方法、装置、设备及存储介质。

【背景技术】

[0002] 无人驾驶车辆，是指通过各种传感器来感知车辆周围环境，并根据感知所获得的道路、车辆位置和障碍物信息等，控制车辆的转向和速度，从而使车辆能够安全、可靠地在道路上行驶。

[0003] 激光雷达是无人驾驶车辆用于感知三维环境的重要传感器，激光雷达扫描一圈场景，返回场景三维空间的点云，即3D点云，包括每个点的三维坐标以及激光的反射强度等。

[0004] 基于采集到3D点云，可进行障碍物的检测等，并反馈给规划控制系统，以便进行避障操作等。可见，障碍物的检测直接关系到无人驾驶车辆的行车安全，具有非常重要的意义。障碍物可包括出现在道路上的行人、机动车、自行车等。

[0005] 现有技术中，主要采用以下方式进行障碍物的检测：首先，采集大量的3D点云，并标注出所有障碍物样本，之后，利用这些样本学习/训练出一个模型，当需要进行实际的检测时，将采集到的3D点云输入该模型，得到障碍物的检测结果。

[0006] 但是，上述方式在实际应用中会存在一定的问题，如：前期需要从大量的3D点云中标注出障碍物样本，用于训练模型，费时费力，增大了工作量，而且，实际使用时，可能会遇到训练时从未遇到过的障碍物，从而导致该障碍物无法被检测出来，即造成障碍物漏检，从而降低了检测结果的准确性。

【发明内容】

[0007] 有鉴于此，本发明提供了障碍物检测方法、装置、设备及存储介质，能够减少工作量和提高检测结果的准确性。

[0008] 具体技术方案如下：

[0009] 一种障碍物检测方法，包括：

[0010] 获取无人驾驶车辆行驶过程中所采集的3D点云；

[0011] 根据所述3D点云分别确定出顶点集及边集；

[0012] 根据所述边集对所述顶点集中的顶点进行聚类，得到最小生成树，作为障碍物检测结果。

[0013] 一种障碍物检测装置，包括：获取单元及检测单元；

[0014] 所述获取单元，用于获取无人驾驶车辆行驶过程中所采集的3D点云，并发送给所述检测单元；

[0015] 所述检测单元，用于根据所述3D点云分别确定出顶点集及边集，并根据所述边集对所述顶点集中的顶点进行聚类，得到最小生成树，作为障碍物检测结果。

[0016] 一种计算机设备，包括存储器、处理器及存储在所述存储器上并可在所述处理器

上运行的计算机程序,所述处理器执行所述程序时实现如以上所述的方法。

[0017] 一种计算机可读存储介质,其上存储有计算机程序,所述程序被处理器执行时实现如以上所述的方法。

[0018] 基于上述介绍可以看出,采用本发明所述方案,不需要进行样本标注及模型训练等,从而相比于现有技术减少了工作量,而且,可适用于对任意类型的障碍物进行检测,从而减少了漏检现象的发生,进而相比于现有技术提高了检测结果的准确性。

【附图说明】

[0019] 图1为本发明所述障碍物检测方法实施例的流程图。

[0020] 图2为本发明所述3D网格的示意图。

[0021] 图3为本发明所述将3D点云投影到与地平面平行的二维网格上的方式示意图。

[0022] 图4为本发明所述3D点云的示意图。

[0023] 图5为本发明所述高度图的示意图。

[0024] 图6为本发明所述障碍物检测装置实施例的组成结构示意图。

[0025] 图7示出了适于用来实现本发明实施方式的示例性计算机系统/服务器12的框图。

【具体实施方式】

[0026] 一般来说,同一个物体上,点与点之间的距离较小,不同物体上,点与点之间的距离较大,因此,可以利用点与点之间的距离来区分不同的物体,本发明中即利用上述思想来进行障碍物的检测。

[0027] 为了使本发明的技术方案更加清楚、明白,以下参照附图并举实施例,对本发明所述方案作进一步地详细说明。

[0028] 图1为本发明所述障碍物检测方法实施例的流程图,如图1所示,包括以下具体实现方式:

[0029] 在101中,获取无人驾驶车辆行驶过程中所采集的3D点云;

[0030] 在102中,根据获取到的3D点云分别确定出顶点集及边集;

[0031] 在103中,根据边集对顶点集中的顶点进行聚类,得到最小生成树,作为障碍物检测结果。

[0032] 在无人驾驶车辆的行驶过程中,激光雷达会不断地进行数据采集,当每次获取到激光雷达返回的3D点云之后,即可按上述102和103所述方式进行处理,即进行障碍物的检测。

[0033] 具体地,本发明中提供了三种障碍物的检测方式,以下分别进行介绍。

[0034] 一)方式一

[0035] 该方式中,可将3D点云中的每个点分别作为一个顶点,利用所有顶点组成顶点集。

[0036] 并且,针对每个顶点,可分别确定出位于该顶点周围预定范围(如预定半径)内的各顶点,并分别计算出确定出的每个顶点与该顶点之间的欧式距离。

[0037] 之后,针对每个欧式距离,可分别将该欧式距离对应的两个顶点之间的连线作为一条边,并将每条边对应的欧式距离作为该条边的长度,利用所有边组成边集。

[0038] 所述预定范围的具体取值可根据实际需要而定。

[0039] 比如,针对顶点 x ,可分别采用kdtree方式搜寻出其周围1米范围内的各顶点,假设为顶点 a 、顶点 b 和顶点 c ,分别计算顶点 a 与顶点 x 、顶点 b 与顶点 x 以及顶点 c 与顶点 x 之间的欧式距离,从而共得到3个欧式距离。

[0040] 针对每个欧式距离,分别将其对应的两个顶点之间的连线作为一条边,即分别将顶点 a 与顶点 x 、顶点 b 与顶点 x 以及顶点 c 与顶点 x 之间的连线作为一条边,从而共得到3条边,所述连线通常为直线。

[0041] 这样,按照上述方式,可针对每个顶点分别得到一条或多条边,所有的边共同组成所述边集。

[0042] 之后,可将各孤立的顶点聚合成若干类,每个类即为一个障碍物,期望的是同一个障碍物上的点聚到一起,而不同障碍物上的点分开。

[0043] 基于上述思想,通常的处理方式为:借用区域增长的思想,先随机选取种子点,然后把周围距离小于阈值的点聚成类,一圈一圈扩散出去,从而得到多个类。

[0044] 但是,这种方式中采用固定阈值,鲁棒性较差,因为近处的障碍物上的点云密集,而远处的障碍物上的点云稀疏,因此,不同距离上的障碍物,不能用同一个阈值去聚类。

[0045] 为此,本发明中提出了一种改进后的算法,可简称为MstSegmentation算法。

[0046] 具体地,初始状态下,可分别将顶点集中的每个顶点作为一个类;之后,可将边集中的各条边按照长度从小大小的顺序进行排序,并按照排序后由先到后的顺序,依次对各条边进行以下处理:确定该边是否满足填加条件,如果是,则将该边对应的两个顶点相连(如通过直线相连),进行类的聚合,即将该边填加到图中,每填加一条边,就会将两个类聚合成一个类;这样,当边集中的所有条边均处理完成后,即可得到聚合出的最小生成树,每棵数就是一类。

[0047] 其中,针对每条边,确定其是否满足填加条件的方式可为:分别计算出该边对应的两个顶点所在的类的阈值;如果该边的长度同时小于计算出的两个阈值,且,如果将该边对应的两个顶点相连后不会出现环,则确定该边满足填加条件。

[0048] 比如,边 x' 对应的两个顶点分别为顶点 a' 和顶点 b' ,需要分别计算出顶点 a' 所在的类的阈值以及顶点 b' 所在的类的阈值,边 x' 的长度即为顶点 a' 和顶点 b' 之间的欧式距离的值。

[0049] 其中,针对顶点 a' ,可首先确定出顶点 a' 所在的类中包括的顶点数Num,并确定出顶点 a' 所在的类中长度最大的边的长度 L ,之后,可用预先设定的常数 K 除以Num,并将得到的商与 L 相加,将相加之和作为顶点 a' 所在的类的阈值。

[0050] 同样地,针对顶点 b' ,可首先确定出顶点 b' 所在的类中包括的顶点数Num,并确定出顶点 b' 所在的类中长度最大的边的长度 L ,之后,可用预先设定的常数 K 除以Num,并将得到的商与 L 相加,将相加之和作为顶点 b' 所在的类的阈值。

[0051] 如果边 x' 的长度同时小于顶点 a' 所在的类的阈值以及顶点 b' 所在的类的阈值,而且,将顶点 a' 和顶点 b' 相连后,不会出现环,则确定边 x' 满足填加条件,相应地,将顶点 a' 和顶点 b' 相连,从而将顶点 a' 所在的类与顶点 b' 所在的类聚合成一个类。

[0052] 上述 K 的取值可根据实际需要而定。

[0053] 针对边集中的每条边,可分别按照上述边 x' 的方式进行处理,直到将边集中的所有边均处理完毕,从而可得到一个或多个最小生成树。

[0054] 可以看出,上述处理过程中的阈值是在不断变化的,即不是固定的,而是根据每个类局部信息(长度最大的边的长度以及顶点数)计算出来的,从而相比于采用固定阈值的方式具有更好的鲁棒性。

[0055] 比如,远处的障碍物上的点间距大,那么L的取值则比较大,而且,远处的障碍物上的点数量少,那么Num的取值则比较小,反之,K/Num则比较大,相应地,远处的障碍物对应的阈值则比较大。

[0056] 二) 方式二

[0057] 在实际应用中,每次采集到的3D点云中包括的点数会非常多,比如,当前广泛使用的velodyne64激光雷达,每次大约能采集到13w个点,如果针对这13w个点,均按照方式一中所述的方式确定顶点集和边集,尤其是确定边集,将会非常耗时,从而降低了处理速度。

[0058] 为此,方式二中提出了一种加速策略,即对3D点云进行下采样,根据下采样结果确定出顶点集及边集。

[0059] 具体地,可首先将3D点云投影到3D网格上,如图2所示,图2为本发明所述3D网格的示意图,网格的具体数量可根据实际需要而定,每个网格记录该网格内所有点的均值。

[0060] 也就是说,针对每个网格,可分别计算出该网格内的各点的均值,并将所述均值对应的点作为一个顶点,利用所有顶点组成顶点集。

[0061] 进一步地,针对每个顶点,可分别确定出位于该顶点周围预定范围内的各顶点,并分别计算出确定出的各顶点与该顶点之间的欧式距离。

[0062] 并且,针对每个欧式距离,可分别将该欧式距离对应的两个顶点之间的连线作为一条边,并将每条边对应的欧式距离作为边的长度,利用所有边组成边集。

[0063] 在得到顶点集和边集之后,即可按照方式一中所述的MstSegmentation算法,根据边集对顶点集中的顶点进行聚类,得到最小生成树,作为障碍物检测结果,具体实现请参照方式一中的相关说明,不再赘述。

[0064] 三) 方式三

[0065] 如前所述,在实际应用中,每次采集到的3D点云中包括的点数会非常多,如13w个点,如果针对这13w个点,均按照方式一中所述的方式确定顶点集及边集,尤其是确定边集,将会非常耗时,从而降低了处理速度。

[0066] 为此,方式三中提出了另外一种加速策略,即根据3D点云生成高度图,根据高度图确定出顶点集及边集。

[0067] 具体地,可将3D点云投影到与地平面平行的二维网格上,图3为本发明所述将3D点云投影到与地平面平行的二维网格上的方式示意图。

[0068] 如图3所示,针对每个网格,可分别计算出该网格对应的柱状体中的各点在x方向和y方向上的均值,并选出该网格对应的柱状体中的各点在z方向上的最大值,将均值对应的点作为一个像素点,将该最大值作为该像素点对应的高度值。

[0069] 也就是说,每个网格记录了对应的柱状体中的各点在x方向和y方向上的均值及z方向上的最大值,理论上,z方向上可以使用最大值、均值和中值等,但测试显示,使用最大值的效果最好,即将3D点云投影到2D图片上,这个图片记录了z方向上的最大值,也可称为最高值,因此可将该图片称为高度图。

[0070] 如图4和图5所示,图4为本发明所述3D点云的示意图,图5为本发明所述高度图的

示意图,图5中的各像素点的灰度值与高度值成正比。

[0071] 可将每个像素点分别作为一个顶点,利用所有顶点组成顶点集。

[0072] 进一步地,针对每个顶点,可分别利用x方向和y方向上的取值以及高度值,计算出该顶点的 $N*N$ 邻域内的其他各顶点与该顶点之间的欧式距离, N 为大于1的奇数,具体取值可根据实际需要而定。

[0073] 比如, N 的取值为3,针对该 $3*3$ 邻域中的中心点,可分别计算除该中心点之外的其他各点与该中心点之间的欧式距离。

[0074] 并且,针对每个欧式距离,可分别将该欧式距离对应的两个顶点之间的连线作为一条边,并将每条边对应的欧式距离作为该条边的长度,利用所有边组成边集。

[0075] 在得到顶点集和边集之后,即可按照方式一中所述的MstSegmentation算法,根据边集对顶点集中的顶点进行聚类,得到最小生成树,作为障碍物检测结果,具体实现请参照方式一中的相关说明,不再赘述。

[0076] 由于是在2D空间中得到的检测结果,因此,还需要将检测结果反投影到3D空间中,具体实现为现有技术。

[0077] 以上共介绍了方式一、方式二和方式三三种障碍物检测方式,三者检测的准确性方面是一样的,但在处理速度上存在着较大的差别,其中的方式二和方式三均能大幅提升顶点集和边集的获取速度,从而提升了整个障碍物检测过程的处理速度。

[0078] 测试显示,针对每次获取到的3D点云,方式一的处理速度为1000ms,方式二的处理速度为60ms,而方式三的处理速度则可达到8ms,处理速度越快,越能满足无人驾驶车辆对于实时性的要求。

[0079] 另外,相比于现有技术,采用本发明所述障碍物检测方式,不需要进行样本标注及模型训练等,从而减少了工作量,而且,可适用于对任意类型的障碍物进行检测,从而减少了漏检现象的发生,进而提高了检测结果的准确性,另外,在实际应用中,可将本发明所述障碍物检测方式与现有障碍物检测方式相结合,以便进一步提高检测结果的准确性等。

[0080] 以上是关于方法实施例的介绍,以下通过装置实施例,对本发明所述方案进行进一步说明。

[0081] 图6为本发明所述障碍物检测装置实施例的组成结构示意图,包括:获取单元601和检测单元602。

[0082] 获取单元601,用于获取无人驾驶车辆行驶过程中所采集的3D点云,并发送给检测单元602。

[0083] 检测单元602,用于根据获取到的3D点云分别确定出顶点集及边集,并根据所述边集对顶点集中的顶点进行聚类,得到最小生成树,作为障碍物检测结果。

[0084] 如图6所示,检测单元602中可具体包括:第一处理子单元6021以及第二处理子单元6022。

[0085] 其中,第一处理子单元6021用于根据获取到的3D点云分别确定出顶点集及边集,可以有三种不同的实现方式。

[0086] 比如,第一处理子单元6021可将获取到的3D点云中的每个点分别作为一个顶点,利用所有顶点组成顶点集;针对每个顶点,分别确定出位于该顶点周围预定范围内的各顶点,并分别计算出确定出的各顶点与该顶点之间的欧式距离;针对每个欧式距离,分别将该

欧式距离对应的两个顶点之间的连线作为一条边,并将每条边对应的欧式距离作为边的长度,利用所有边组成边集。

[0087] 或者,第一处理子单元6021可对获取到的3D点云进行下采样,根据下采样结果确定出顶点集及边集。

[0088] 具体地,第一处理子单元6021可进行以下处理:

[0089] 将3D点云投影到3D网格上;

[0090] 针对每个网格,分别计算出该网格中的各点的均值,并将均值对应的点作为一个顶点,利用所有顶点组成顶点集;

[0091] 针对每个顶点,分别确定出位于该顶点周围预定范围内的各顶点,并分别计算出确定出的各顶点与该顶点之间的欧式距离;

[0092] 针对每个欧式距离,分别将该欧式距离对应的两个顶点之间的连线作为一条边,并将每条边对应的欧式距离作为边的长度,利用所有边组成边集。

[0093] 又或者,第一处理子单元6021可根据获取到的3D点云生成高度图,根据高度图确定出顶点集及边集。

[0094] 具体地,第一处理子单元6021可进行以下处理:

[0095] 将3D点云投影到与地平面平行的二维网格上;

[0096] 针对每个网格,分别计算出该网格对应的柱状体中的各点在x方向和y方向上的均值,并选出该网格对应的柱状体中的各点在z方向上的最大值,将均值对应的点作为一个像素点,将该最大值作为该像素点对应的高度值,得到高度图;

[0097] 将每个像素点分别作为一个顶点,利用所有顶点组成顶点集;

[0098] 针对每个顶点,分别利用x方向和y方向上的取值以及高度值,计算出该顶点的N*N邻域内的其他各顶点与该顶点之间的欧式距离,N为大于1的奇数;

[0099] 针对每个欧式距离,分别将该欧式距离对应的两个顶点之间的连线作为一条边,并将每条边对应的欧式距离作为边的长度,利用所有边组成边集。

[0100] 无论是采用上述哪种方式,在确定出顶点集及边集之后,第二处理子单元6022即可进行以下处理:

[0101] 在初始状态下,分别将顶点集中的每个顶点作为一个类;

[0102] 将边集中的各条边按照长度从小到大的顺序进行排序,并按照排序后由先到后的顺序,依次对各条边进行以下处理:确定该边是否满足添加条件,如果是,则将该边对应的两个顶点相连,进行类的聚合;

[0103] 当边集中的各条边均处理完成后,得到聚合出的最小生成树。

[0104] 其中,满足添加条件可以是指:边的长度同时小于该边对应的两个顶点所在的类的阈值,且,将该边对应的两个顶点相连后不会出现环。

[0105] 每个顶点所在的类的阈值= $L+K/Num$;

[0106] 其中,Num表示该顶点所在的类中包括的顶点数;

[0107] L表示该顶点所在的类中长度最大的边的长度;

[0108] K表示预先设定的常数。

[0109] 图6所示装置实施例的具体工作流程请参照前述方法实施例中的相应说明,不再赘述。

[0110] 图7示出了适于用来实现本发明实施方式的示例性计算机系统/服务器12的框图。图7显示的计算机系统/服务器12仅仅是一个示例,不应对本发明实施例的功能和使用范围带来任何限制。

[0111] 如图7所示,计算机系统/服务器12以通用计算设备的形式表现。计算机系统/服务器12的组件可以包括但不限于:一个或者多个处理器(处理单元)16,存储器28,连接不同系统组件(包括存储器28和处理器16)的总线18。

[0112] 总线18表示几类总线结构中的一种或多种,包括存储器总线或者存储器控制器,外围总线,图形加速端口,处理器或者使用多种总线结构中的任意总线结构的局域总线。举例来说,这些体系结构包括但不限于工业标准体系结构(ISA)总线,微通道体系结构(MAC)总线,增强型ISA总线、视频电子标准协会(VESA)局域总线以及外围组件互连(PCI)总线。

[0113] 计算机系统/服务器12典型地包括多种计算机系统可读介质。这些介质可以是任何能够被计算机系统/服务器12访问的可用介质,包括易失性和非易失性介质,可移动的和不可移动的介质。

[0114] 存储器28可以包括易失性存储器形式的计算机系统可读介质,例如随机存取存储器(RAM)30和/或高速缓存存储器32。计算机系统/服务器12可以进一步包括其它可移动/不可移动的、易失性/非易失性计算机系统存储介质。仅作为举例,存储系统34可以用于读写不可移动的、非易失性磁介质(图7未显示,通常称为“硬盘驱动器”)。尽管图7中未示出,可以提供用于对可移动非易失性磁盘(例如“软盘”)读写的磁盘驱动器,以及对可移动非易失性光盘(例如CD-ROM, DVD-ROM或者其它光介质)读写的光盘驱动器。在这些情况下,每个驱动器可以通过一个或者多个数据介质接口与总线18相连。存储器28可以包括至少一个程序产品,该程序产品具有一组(例如至少一个)程序模块,这些程序模块被配置以执行本发明各实施例的功能。

[0115] 具有一组(至少一个)程序模块42的程序/实用工具40,可以存储在例如存储器28中,这样的程序模块42包括——但不限于——操作系统、一个或者多个应用程序、其它程序模块以及程序数据,这些示例中的每一个或某种组合中可能包括网络环境的实现。程序模块42通常执行本发明所描述的实施例中的功能和/或方法。

[0116] 计算机系统/服务器12也可以与一个或多个外部设备14(例如键盘、指向设备、显示器24等)通信,还可与一个或者多个使得用户能与该计算机系统/服务器12交互的设备通信,和/或与使得该计算机系统/服务器12能与一个或多个其它计算设备进行通信的任何设备(例如网卡,调制解调器等等)通信。这种通信可以通过输入/输出(I/O)接口22进行。并且,计算机系统/服务器12还可以通过网络适配器20与一个或者多个网络(例如局域网(LAN),广域网(WAN)和/或公共网络,例如因特网)通信。如图7所示,网络适配器20通过总线18与计算机系统/服务器12的其它模块通信。应当明白,尽管图中未示出,可以结合计算机系统/服务器12使用其它硬件和/或软件模块,包括但不限于:微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID系统、磁带驱动器以及数据备份存储系统等。

[0117] 处理器16通过运行存储在存储器28中的程序,从而执行各种功能应用以及数据处理,例如实现图1所示实施例中的方法,即获取无人驾驶车辆行驶过程中所采集的3D点云,根据获取到的3D点云分别确定出顶点集及边集,根据所述边集对顶点集中的顶点进行聚类,得到最小生成树,作为障碍物检测结果。

[0118] 其中，“根据获取到的3D点云分别确定出顶点集及边集，根据所述边集对顶点集中的顶点进行聚类，得到最小生成树”可以有之前所述的方式一、方式二和方式三三种实现方式。

[0119] 本发明同时公开了一种计算机可读存储介质，其上存储有计算机程序，该程序被处理器执行时将实现如图1所示实施例中的方法。

[0120] 可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括：具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中，计算机可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0121] 计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号，其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式，包括——但不限于——电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质，该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

[0122] 计算机可读介质上包含的程序代码可以用任何适当的介质传输，包括——但不限于——无线、电线、光缆、RF等等，或者上述的任意合适的组合。

[0123] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机程序代码，所述程序设计语言包括面向对象的程序设计语言——诸如Java、Smalltalk、C++，还包括常规的过程式程序设计语言——诸如“C”语言或类似的设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络——包括局域网(LAN)或广域网(WAN)——连接到用户计算机，或者，可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0124] 在本发明所提供的几个实施例中，应该理解到，所揭露的装置和方法等，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式。

[0125] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0126] 另外，在本发明各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用硬件加软件功能单元的形式实现。

[0127] 上述以软件功能单元的形式实现的集成的单元，可以存储在一个计算机可读取存

储介质中。上述软件功能单元存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备)或处理器(processor)执行本发明各个实施例所述方法的部分步骤。而前述的存储介质包括:U盘、移动硬盘、ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0128] 以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本发明保护的范围之内。

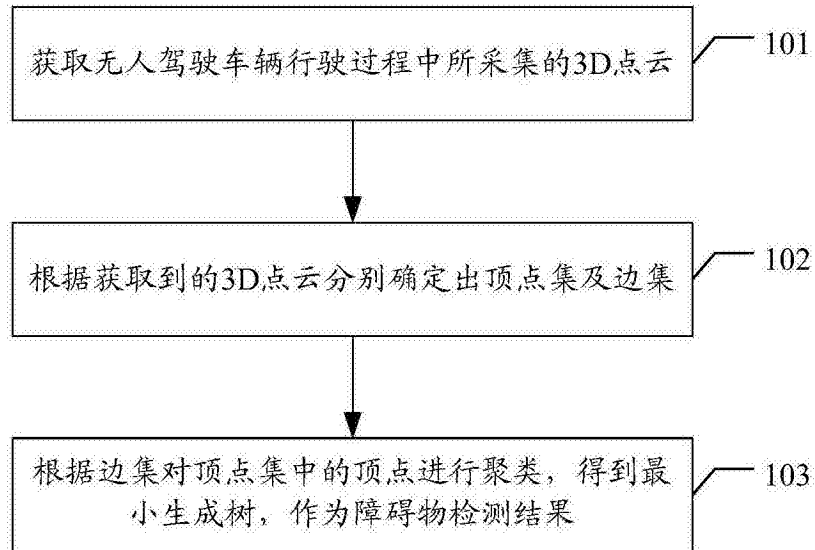


图1

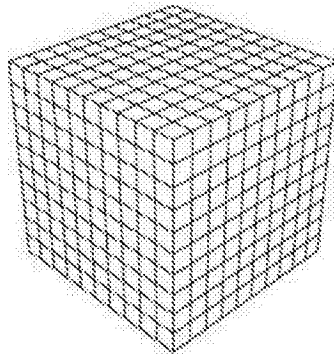


图2

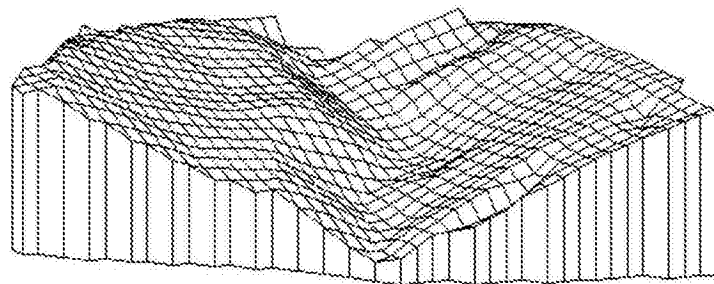


图3

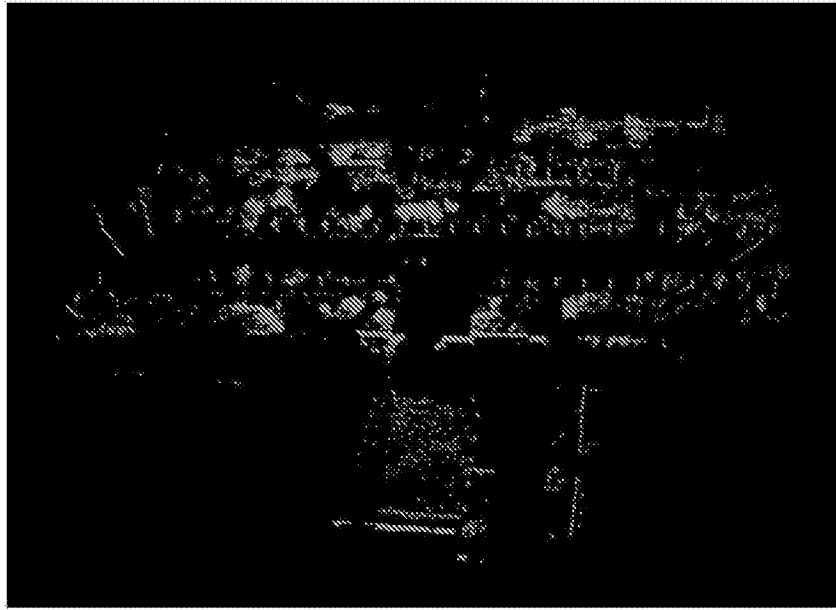


图4

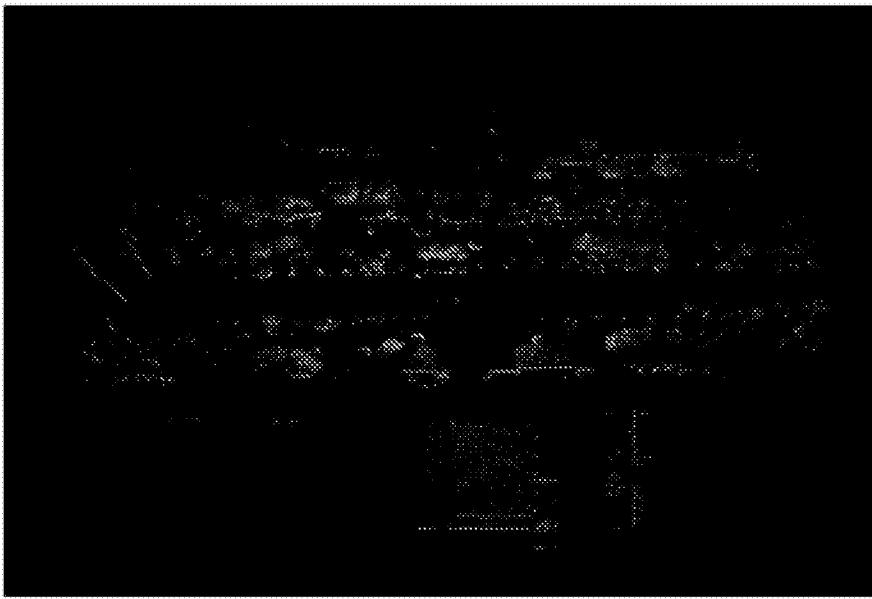


图5

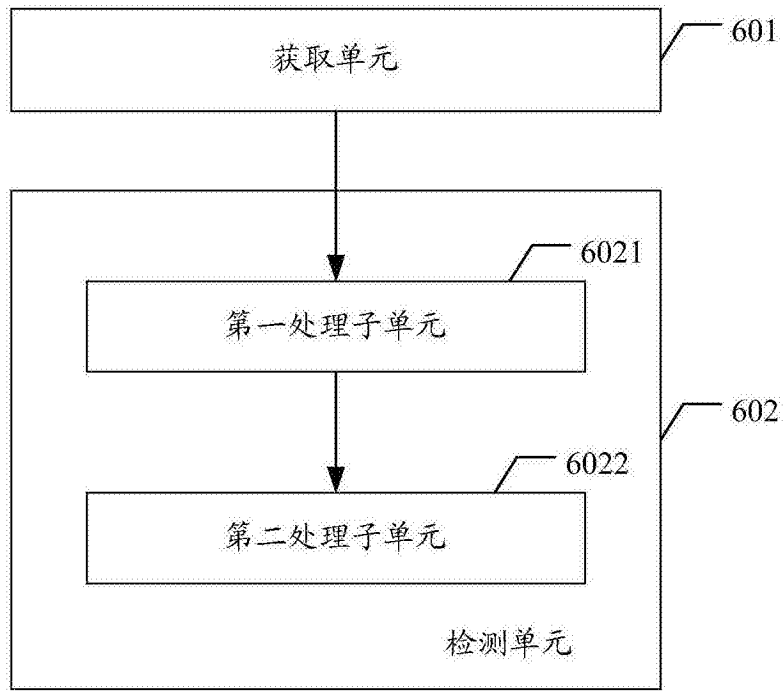


图6

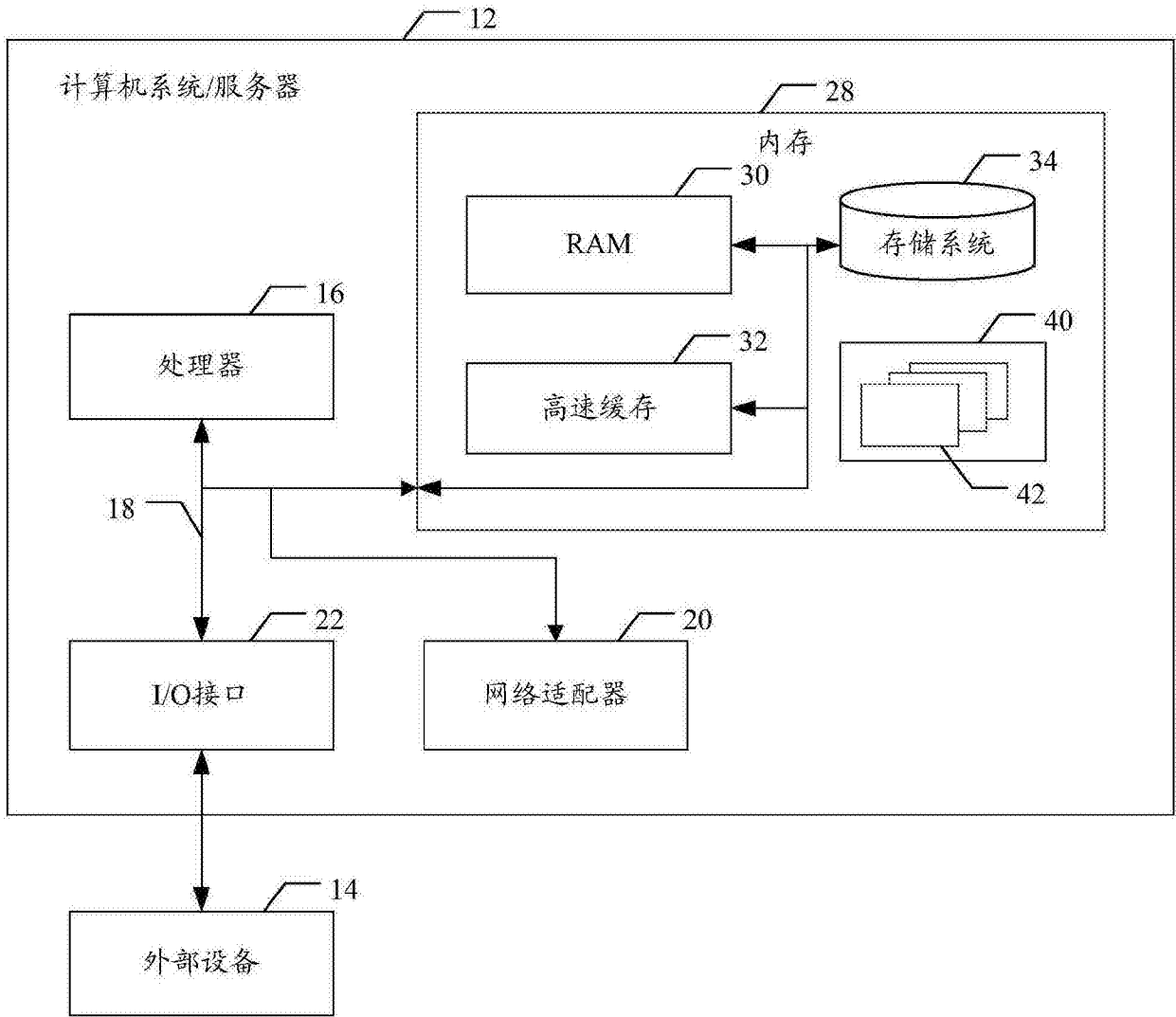


图7