(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112304314 A
(43)申请公布日 2021.02.02

(21)申请号 202011125420.4
(22)申请日 2020.10.20
(66)本国优先权数据
202010914241.2 2020.08.27 CN
(71)申请人 中国科学技术大学
地址 230026 安徽省合肥市包河区金寨路96号
(72)发明人 陈广大 姚舜一 吉建民
(74)专利代理机构 北京凯特来知识产权代理有限公司 11260
代理人 郑立明 付久春
(51)Int.Cl.
G01C 21/20(2006.01)
G06F 30/27(2020.01)
G06N 3/04(2006.01)
G06N 3/08(2006.01)

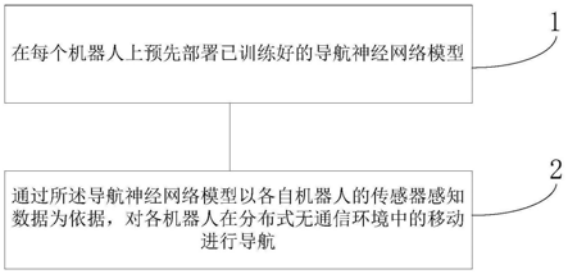
权利要求书2页 说明书9页 附图4页

(54)发明名称

一种分布式多机器人的导航方法

(57)摘要

本发明公开了一种分布式多机器人的导航方法，包括：在每个机器人上预先部署已训练好的导航神经网络模型，通过所述导航神经网络模型以各自机器人的传感器感知数据为依据，对各机器人在分布式无通信环境中的移动进行导航；所述导航神经网络模型的训练是在仿真训练场景中，依据机器人的传感器感知数据形成的栅格地图进行深度学习的仿真移动训练。该方法在机器人上部署深度强化学习训练的导航神经网络模型，由多种传感器信息生成栅格地图，以机器人为中心的局部栅格地图作为网络输入，具有抗噪音能力强，计算代价低等优点。结合全局路径规划器，能在分布式无通信的环境中对机器人无碰撞移动导航，可应用在多种机器人导航的场景。



CN 112304314 A

1.一种分布式多机器人的导航方法,其特征在于,用于在设有至少两个机器人的分布式无通信环境中对各机器人的移动进行导航,包括以下步骤:

在每个机器人上预先部署已训练好的导航神经网络模型,通过所述导航神经网络模型以各自机器人的传感器感知数据为依据,对各机器人在分布式无通信环境中的移动进行导航;

所述导航神经网络模型的训练是在仿真训练场景中,依据机器人的传感器感知数据形成的栅格地图进行深度学习的仿真移动训练。

2.根据权利要求1所述的分布式多机器人的导航方法,其特征在于,所述导航神经网络模型包括:

策略网络和值网络;其中,

所述策略网络包括三个2D卷积层,每个2D卷积层后设置一个最大池化层,三个2D卷积层之后设置全连接层;其中,第一个2D卷积层为64维,3x3的卷积核和ReLU激活函数;第二个2D卷积层为128维,3x3的卷积核和ReLU激活函数;第三个2D卷积层为256维,3x3的卷积核和ReLU激活函数;

每个最大池化层的核为2x2,步长为2;

所述全连接层为512单位的全连接层,所述最后一层全连接层输出角速度和线速度的均值,随机采样出要执行的动作;

所述值网络的结构与所述策略网络的结构相同,该值网络的最后一层全连接为一维,不含激活函数,输出值为函数值;所述值网络的输出值用于策略网络计算梯度更新参数。

3.根据权利要求1或2所述的分布式多机器人的导航方法,其特征在于,所述导航神经网络模型的训练是在仿真训练场景中,依据机器人的传感器感知数据形成的栅格地图进行深度学习的仿真移动训练为:

步骤S1.采用仿真软件读入预先生成的地图文件和描述场景信息的配置文件生成随机的仿真训练场景,所述仿真训练场景中设有不同形状障碍物和机器人;

步骤S2.在所述步骤S1生成的仿真训练场景中,依据机器人的传感器感知数据形成的栅格地图对机器人的导航神经网络模型进行训练并收集经验数据;

步骤S3.使用收集的经验数据训练更新导航神经网络模型的参数,若判断导航神经网络模型参数未收敛,则返回至步骤S1进行迭代训练,直至所述导航神经网络模型参数收敛。

4.根据权利要求3所述的分布式多机器人的导航方法,其特征在于,所述步骤S1的仿真软件中机器人的运动学模型为:$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v}{\omega}\sin\theta + \frac{v}{\omega}\sin(\theta+\omega\Delta t) \\ \frac{v}{\omega}\cos\theta - \frac{v}{\omega}\cos(\theta+\omega\Delta t) \\ \omega\Delta t \end{pmatrix};$$

上述运动学模型中,x'y'θ'为下一时刻机器人在全局坐标系下的位置;xyθ为前一时刻机器人在全局坐标系下的位置;v为机器人的线速度;ω为机器人的角速度;Δt为机器人每次运动的时间间隔;

所述仿真训练场景中,状态空间包含:三帧由传感器数据和机器人形状生成的栅格地图图像,三帧目标点相对机器人坐标系的位置和三帧机器人与起始位姿的偏角;在栅格地图中,格子值为255的区域表示无障碍物空白区域,格子值为100的区域表示未知区域,格子

值为0的区域表示障碍物区域，格子值为200的区域表示机器人体积形状区域；

对栅格地图的每个值除以255归一化得到栅格图像；

动作空间定义为机器人的角速度和线速度；

所述机器人的导航神经网络模型的回报函数rt为：rt＝rtg+rta+rtc+rts；

其中，rtg为诱导到达值，rtg＝200(|pt-pg|-|pt1-pg|)；

rta为t时刻到达奖励值，当机器人到达目标点时，则rta等于500，否则rta等于0，当pt与pg距离小于0.20为机器人到达目标点；

rtc为t时刻碰撞惩罚值，如果t时刻机器人发生碰撞，则rtc等于-500，否则rtc等于0；

rts为t时刻固定惩罚值，rts设置为-5；

所述pg表示机器人的目标点，pt表示t时刻机器人的位置，pt1表示t+1时刻机器人的位置，|pt-pg|表示pt和pg位置之间的直线距离。

5.根据权利要求3所述的分布式多机器人的导航方法，其特征在于，所述步骤S3中，使用收集的经验数据训练更新导航神经网络模型的参数为：基于分布式近端策略优化算法更新所述导航神经网络模型的参数。

6.根据权利要求3所述的分布式多机器人的导航方法，其特征在于，所述步骤S1中，构建的仿真训练场景为不同构成的多个随机环境。

7.根据权利要求4所述的分布式多机器人的导航方法，其特征在于，所述多个随机环境包括：

第一种随机仿真环境为含有8个机器人和4个障碍物的随机环境，其中，机器人的起点和终点在每回合都随机生成，终点距离起点2m到4m远；障碍物包括：两个矩形和两个圆形，障碍物的尺寸会随机调整；

第二种仿真环境为含有8个机器人的随机圆形环境，其中，机器人起点在圆上随机，终点位于正对起点的圆上，每回合圆的半径在1.8到3米之间随机调整。

一种分布式多机器人的导航方法

[0001] 本申请要求于2020年8月27日提交中国专利局、申请号为202010914241.2、发明名称为“一种基于格栅地图和深度强化学习的分布式多机器人导航方法”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

[0002] 本发明涉及机器人导航领域，尤其涉及一种分布式多机器人的导航方法。

背景技术

[0003] 目前，多机器人避障技术可以分为两大类：中心化的方法和去中心化的方法。中心化的方法往往需要一个中心服务器来统一规划各个机器人下一步的动作指令，因此需要各个机器人与中心服务器进行通讯，同时随着机器人的增多，系统的通信代价加大，稳定性降低。不同于中心化的方法，去中心化的方法是让机器人根据自身局部观察，各自自主规划下一步动作指令，去中心化的方法通常可分为两大类：智能体级别(agent-level)和传感器级别(sensor-level)；其中，智能体级别的方法是根据感知到的周围机器人的位置和速度等语义信息来决策，但此类方法需要高计算代价的前端感知模块处理，使整个系统效果和稳定性降低，并且增加了计算代价；而传感器级别的方法是根据传感器的原始数据信息来决策，此类方法不需要复杂的前端感知模块，但这类方法不能兼容不同传感器配置，部署受限。

[0004] 如上所述，现有的多机器人在分布式无通信的环境中，根据自身传感器的部分观察进行安全高效地无碰撞移动导航是非常困难的。

发明内容

[0005] 基于现有技术所存在的问题，本发明的目的是提供一种分布式多机器人的导航方法，能解决现有多机器人在分布式无通信的环境中，根据自身传感器的部分观察不能实现安全高效地无碰撞移动导航的问题。

[0006] 本发明的目的是通过以下技术方案实现的：

[0007] 本发明实施方式提供一种分布式多机器人的导航方法，用于在设有至少两个机器人的分布式无通信环境中对各机器人的移动进行导航，包括以下步骤：

[0008] 在每个机器人上预先部署已训练好的导航神经网络模型，通过所述导航神经网络模型以各自机器人的传感器感知数据为依据，对各机器人在分布式无通信环境中的移动进行导航；

[0009] 所述导航神经网络模型的训练是在仿真训练场景中，依据机器人的传感器感知数据形成的栅格地图进行深度学习的仿真移动训练。

[0010] 由上述本发明提供的技术方案可以看出，本发明实施例提供的分布式多机器人的导航方法，其有益效果为：

[0011] 通过在机器人上部署深度强化学习训练的导航神经网络模型，由多种传感器信息

生成栅格地图，以机器人为中心的局部栅格地图作为网络输入，具有抗噪音能力强，计算代价低等优点。结合全局路径规划器，能在分布式无通信的环境中对机器人无碰撞移动导航，可应用在多种机器人导航的场景。

附图说明

[0012] 为了更清楚地说明本发明实施例的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域的普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他附图。

[0013] 图1为本发明实施例提供的分布式多机器人的导航方法的流程图；

[0014] 图2为本发明实施例提供的分布式多机器人的导航方法的训练步骤流程图；

[0015] 图3为本发明实施例提供的导航神经网络模型的策略网络的构成示意图；

[0016] 图4为本发明实施例提供的分布式多机器人的导航方法在多种仿真测试环境的路径轨迹图；其中，(a)至(d)为四种圆形环境的仿真训练场景的示意图；(e)为一种交叉环境的仿真训练场景的示意图；(f)为一种交换环境的仿真训练场景的示意图；(g)和(h)为一种随机环境的仿真训练场景的示意图；

[0017] 图5为本发明实施例提供的分布式多机器人的导航方法的短距离导航中，单个机器人在静态障碍物场景的路径轨迹图；

[0018] 图6为本发明实施例提供的分布式多机器人的导航方法的短距离导航中，多个机器人在有静态障碍物和动态行人场景的路径轨迹图；

[0019] 图7为本发明实施例提供的分布式多机器人的导航方法的长距离导航中机器人的整体路径图；

[0020] 图8为本发明实施例提供的分布式多机器人的导航方法的仿真环境示意图；其中，(a)为第一种随机环境的仿真训练场景示意图，(b)第二种随机环境的仿真训练场景示意图。

具体实施方式

[0021] 下面结合本发明的具体内容，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明的保护范围。本发明实施例中未作详细描述的内容属于本领域专业技术人员公知的现有技术。

[0022] 如图1所示，本发明实施例提供的一种分布式多机器人的导航方法，用于在设有至少两个机器人的分布式无通信环境中对各机器人的移动进行导航，包括以下步骤：

[0023] 在每个机器人上预先部署已训练好的导航神经网络模型，通过所述导航神经网络模型以各自机器人的传感器感知数据为依据，对各机器人在分布式无通信环境中的移动进行导航；

[0024] 所述导航神经网络模型的训练是在仿真训练场景中，依据机器人的传感器感知数据形成的栅格地图进行深度学习的仿真移动训练。

[0025] 上述导航方法中，所述导航神经网络模型包括：

[0026] 策略网络和值网络；其中，

[0027] 如图3所示，所述策略网络包括三个2D卷积层，每个2D卷积层后设置一个最大池化层，三个2D卷积层之后设置全连接层；其中，第一个2D卷积层为64维，3x3的卷积核和ReLU激活函数；第二个2D卷积层为128维，3x3的卷积核和ReLU激活函数；第三个2D卷积层为256维，3x3的卷积核和ReLU激活函数；

[0028] 每个最大池化层的核为2x2，步长为2；

[0029] 所述全连接层为512单位的全连接层，所述最后一层全连接层输出角速度和线速度的均值，随机采样出要执行的动作；

[0030] 所述值网络的结构与所述策略网络的结构相同，该值网络的最后一层全连接为一维，不含激活函数，输出值为函数值；

[0031] 所述值网络的输出值用于所述策略网络计算梯度更新参数。

[0032] 如图2所示，上述导航方法中，所述导航神经网络模型的训练是在仿真训练场景中，依据机器人的传感器感知数据形成的栅格地图进行深度学习的仿真移动训练为：

[0033] 步骤S1.采用仿真软件读入预先生成的地图文件和描述场景信息的配置文件生成随机的仿真训练场景，所述仿真训练场景中设有不同形状障碍物和机器人；

[0034] 步骤S2.在所述步骤S1生成的仿真训练场景中，依据机器人的传感器感知数据形成的栅格地图对机器人的导航神经网络模型进行训练并收集经验数据；

[0035] 步骤S3.使用收集的经验数据训练更新导航神经网络模型的参数，若判断导航神经网络模型参数未收敛(导航神经网络模型的期望回报值还在增大)，则返回至步骤S1进行迭代训练，直至所述导航神经网络模型参数收敛(导航神经网络模型的期望回报值不在增大，期望回报值形成的训练曲线不再上升)。

[0036] 上述导航方法中，所述步骤S1的仿真软件中机器人的运动学模型为：

[0037] $$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v}{\omega}\sin\theta + \frac{v}{\omega}\sin(\theta+\omega\Delta t) \\ \frac{v}{\omega}\cos\theta - \frac{v}{\omega}\cos(\theta+\omega\Delta t) \\ \omega\Delta t \end{pmatrix};$$

[0038] 上述运动学模型中，x'y'θ'为下一时刻机器人在全局坐标系下的位置；xyθ为前一时刻机器人在全局坐标系下的位置；v为机器人的线速度；ω为机器人的角速度；Δt为机器人每次运动的时间间隔；

[0039] 所述仿真训练场景中，状态空间包含：三帧由传感器数据和机器人形状生成的栅格地图图像，三帧目标点相对机器人坐标系的位置和三帧机器人与起始位姿的偏角；在栅格地图中，格子值为255的区域表示无障碍物空白区域，格子值为100的区域表示未知区域，格子值为0的区域表示障碍物区域，格子值为200的区域表示机器人体积形状区域；

[0040] 对栅格地图的每个值除以255归一化得到栅格图像；

[0041] 动作空间定义为机器人的角速度和线速度；

[0042] 所述机器人的导航神经网络模型的回报函数rt为：rt＝rtg+rta+rtc+rts；

[0043] 其中，rtg为诱导到达值，rtg＝200(|pt-pg|-|pt1-pg|)；

[0044] rta为t时刻到达奖励值，当机器人到达目标点时，则rta等于500，否则rta等于0，

当pt与pg距离小于0.20为机器人到达目标点；

[0045] rtc为t时刻碰撞惩罚值，如果t时刻机器人发生碰撞，则rtc等于-500，否则rtc等于0；

[0046] rts为t时刻固定惩罚值，rts设置为-5；

[0047] 所述pg表示机器人的目标点，pt表示t时刻机器人的位置，pt1表示t+1时刻机器人的位置，|pt-pg|表示pt和pg位置之间的直线距离。

[0048] 上述导航方法的步骤S3中，使用收集的经验数据训练更新导航神经网络模型的参数为：基于分布式近端策略优化算法更新所述导航神经网络模型的参数。

[0049] 上述导航方法的步骤S1中，构建的仿真训练场景为不同构成的多个随机环境。

[0050] 所述多个随机环境包括：

[0051] 如图8(a)所示，第一种随机仿真环境为含有8个机器人和4个障碍物的随机环境，其中，机器人的起点和终点在每回合都随机生成，终点距离起点2m到4m远；障碍物包括：两个矩形和两个圆形，障碍物的尺寸会随机调整；

[0052] 如图8(b)所示，第二种仿真环境为含有8个机器人的随机圆形环境，其中，机器人起点在圆上随机，终点位于正对起点的圆上，每回合圆的半径在1.8到3米之间随机调整。

[0053] 本发明的导航方法，使多个机器人能在分布式无通信的环境中无碰撞移动导航，由于机器人用于导航的导航神经网络模型使用栅格地图和深度强化学习在仿真环境中进行训练，模型收敛之后能很容易的从仿真环境中移植到实体环境上，本发明的方法安全，高效，在多种仿真环境和实体环境的测试中表现优异。本发明方法不需要复杂的前端感知模块，根据生成的栅格地图来决策，同时具有兼容不同传感器的优点。

[0054] 下面对本发明实施例具体作进一步地详细描述。

[0055] 如图1所示，本发明实施例提供的分布式多机器人的导航方法，是基于栅格地图和深度强化学习为各机器人在分布式无通信的环境中无碰撞移动导航，包括以下步骤：

[0056] 在每个机器人上预先部署已训练好的导航神经网络模型，通过所述导航神经网络模型以各自机器人的传感器感知数据为依据，对各机器人在分布式无通信环境中的移动进行导航；

[0057] 对每个机器人的导航神经网络模型采用以下方式进行移动导航仿真训练，包括：

[0058] 步骤S1.构建仿真学习环境：

[0059] 采用能支持不同形状的障碍物和机器人的仿真软件，通过仿真软件形成仿真训练场景，仿真软件具体通过读入预先生成的地图文件和描述场景信息的配置文件生成每次的仿真训练场景，仿真训练场景中设有不同形状的障碍物和机器人，形成的仿真训练场景作为机器人的导航神经网络模型训练场景；

[0060] 上述步骤S1中，仿真软件中机器人的运动学模型为：

[0061] $$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v}{\omega}\sin\theta + \frac{v}{\omega}\sin(\theta+\omega\Delta t) \\ \frac{v}{\omega}\cos\theta - \frac{v}{\omega}\cos(\theta+\omega\Delta t) \\ \omega\Delta t \end{pmatrix};$$

[0062] 其中，x'y'θ'为下一时刻机器人的位置，是表示位置坐标x'，y'，θ'；xyθ为前一时刻机器人在全局坐标系下的位置，是表示位置坐标x，y，θ；v为机器人线速度；ω为机器人角

速度；Δt为每次运动的时间间隔，由于仿真软件没有通信延迟以及运动学和传感器的误差，用仿真软件对机器人进行导航的导航神经网络模型进行训练，保证了训练结果的正确性；

[0063] 在构建的仿真训练场景中，机器人的状态动作为：

[0064] 观察状态空间包含三帧由传感器数据和机器人形状生成的栅格地图图像，三帧目标点相对机器人坐标系的位置和三帧机器人与起始位姿的偏角。在栅格地图中，格子值为255的区域表示无障碍物空白区域，格子值为100的区域表示未知区域，格子值为0的区域表示障碍物区域，格子值为200的区域表示机器人体积形状区域。栅格地图的每个值除以255归一化得到栅格图像。动作空间定义为机器人的角速度和线速度；

[0065] 设定所训练的机器人的导航神经网络模型的回报函数rt为：

[0066] rt＝rtg+rta+rtc+rts；

[0067] 其中，rtg为诱导到达项，表示上一步与目标点的距离减去当前与目标点的距离然后乘10，表示越靠近目标点越奖励，越远离目标点越惩罚：rtg＝200(|pt-pg|-|pt1-pg|)；

[0068] rta为t时刻到达奖励值，当机器人到达目标点(此时，pt-pg＜0.20)时奖励值为500，即rta＝500，否则该项为0，即rta＝0；

[0069] rtc为t时刻碰撞惩罚值，如果t时刻机器人发生碰撞则给与-500的惩罚，即rtc＝-500，否则该项为0，即发生碰撞，则rtc＝0；

[0070] rts为t时刻固定惩罚值，设置为-5，用来使机器人尽快的到达目标点；

[0071] 上述公式中pg表示机器人的目标点，pt表示t时刻机器人的位置，pt1表示t+1时刻机器人的位置，|pt-pg|表示pt和pg位置之间的直线距离；

[0072] 上述的回报函数在强化学习中起到“指挥棒”的作用。

[0073] 本实施例的步骤S1中，构建了两种仿真训练场景来训练导航神经网络模型，如图8(a)所示，第一种随机仿真环境为8个机器人和4个障碍物的随机环境，其中机器人的起点和终点在每回合都随机生成，终点距离起点2m到4m远，障碍物包括两个矩形和两个圆形，障碍物的尺寸也会随机改变；如图8(b)所示，第二种随机仿真环境为含有8个机器人的随机圆形环境，机器人起点在圆上随机，终点位于正对起点的圆上，每回合圆的半径在1.8到3米间随机。

[0074] 本步骤S1在每次训练时，能随机生成不同构成的仿真训练场景。

[0075] 步骤S2.按步骤S1构建的仿真训练场景对导航神经网络模型进行仿真训练；

[0076] 导航神经网络模型由值网络和策略网络组成，值网络和策略网络均为深度神经网络，其中，策略网络的结构如图3所示，前三层2D卷积和最大池化用于提取栅格图像信息，第一个卷积层为64维，3x3的卷积核和ReLU激活函数，第二个卷积层为128维，3x3的卷积核和ReLU激活函数，第三个卷积层为256维，3x3的卷积核和ReLU激活函数。其中三个最大池化层的核为2x2，步长为2。图3中细条状层为512单位的全连接层。策略网络最后一层全连接输出角速度和线速度的均值，然后随机采样出要执行的动作。值网络的结构与策略网络相同，但它的最后一层全连接为1维，不含激活函数，输出值函数值。

[0077] 步骤S3.基于分布式近端策略优化算法更新导航神经网络模型参数，若判断导航神经网络模型参数未收敛，则返回至步骤S1进行迭代训练，直至所述导航神经网络模型参数收敛：

[0078] 近端策略优化算法是一种新型的策略梯度算法。由于策略梯度算法对步长十分敏感，但是又难以选择合适的步长，在训练过程中新旧策略的变化差异如果过大则不利于学习。标准的策略梯度算法对每个数据样本只执行一次梯度更新，对数据利用效率较低，近端策略优化算法通过与环境的交互作用来采样数据，并利用随机梯度上升来优化替代目标函数的方法来支持多个回合的小批量更新，解决了策略梯度算法中步长难以确定的问题。

[0079] 上述方法中，多机器人分布式PPO算法的伪代码如下：

[0080]

Algorithm 1: Distributed Proximal Policy Optimization

Initialize policy network π_θ and value function $V_\phi(s_t)$.
for *epoch* $= 1, 2..., M$ **do**
  // Collect data in parallel
  for *step* $t = 1, 2..., T_{ep}$ **do**
    if *all robot finish* **then**
      Estimate advantages using GAE $\hat{A}_i^t = \sum_{l=0}^{T_i} (\gamma\lambda)^l \delta_i^l$, where $\delta_i^l = r_i^l + \gamma V_\phi(s_i^{t+1}) - V_\phi(s_i^t)$.
      $\mathbf{o}^t = reset()$
    end
    for *robot* $i = 1, 2..., N$ **do**
      if *reach max steps* T_m **then**
        Cut trajectory, reset environment.
      else if $\exists k \in [1, M] : (\boldsymbol{p}_{obs} \vee \boldsymbol{p}_{hum})_M \in \Omega(\boldsymbol{p}_t)$ *or* $\left\| \boldsymbol{p}_t - \boldsymbol{p}_g \right\| < 0.2$ **then**
        Finish episode, reset environment.
      else
        $\mathbf{a}_i^t = \pi_\theta(\mathbf{o}_i^t)$
        $\mathbf{o}_{i+1}^t, r_{i+1}^t = step(\mathbf{a}_i^t)$
      end
    end
  end
  $\pi_{old} \leftarrow \pi_\theta$
  // Update policy network
  for $m = 1, ..., E_\pi$ **do**
    $L^{PPO}(\theta) = \sum_{t=1}^{T_{ep}} min(\frac{\pi_\theta(a_i^t|o_i^t)}{\pi_{old}(a_i^t|o_i^t)} \hat{A}_i^t, clip(\frac{\pi_\theta(a_i^t|o_i^t)}{\pi_{old}(a_i^t|o_i^t)}, 1-\varepsilon, 1+\varepsilon)\hat{A}_i^t)$
    if $KL[\pi_{old}|\pi_\theta] > 1.5KL_{target}$ **then**
      break
    end
    Update θ with lr_θ by Adam w.r.t $L^{PPO}(\theta)$.
  end
  // Update value function
  for $n = 1, ..., E_V$ **do**
    $L^V(\phi) = -\sum_N^{i=1} \sum_{t=1}^{T_i} (\sum_{t'>t} \gamma^{t'-t} r_i^{t'} - V_\phi(s_i^t))^2$
    Update ϕ with lr_ϕ by Adam w.r.t $L^V(\phi)$.
  end
end

[0081] 多机器人分布式PPO算法的各项参数设置如下：

[0082]

Parameter	Value
T_{ep} in line 4	2000
λ in line 6	0.95
γ in line 6 and 31	0.99
T_m in line 10	200
E_π in line 22	80
ε in line 23	0.2
KL_{target} in line 23	0.01
lr_θ in line 27	3×10^{-4} (Stage 1), 1×10^{-4} (Stage 1)
E_v in line 30	80
lr_ϕ in line 32	1×10^{-3}

[0083] 本发明的导航方法,采用集中学习、分散执行的方式扩展近端策略优化算法,让不同机器人使用相同的策略网络和参数进行决策,在不同的环境中收集经验,然后定期根据不同机器人在不同环境中收集的经验进行集中训练,很好的解决了多机器人避障的问题。

[0084] 在实际训练过程中,使用课程学习的思想,先学习简单的环境,然后再在复杂的环境中学习,这种训练方式是的策略进一步得到了提升。具体地说,首先在第一种仿真环境(图5(a))中训练,到600回合左右后停止训练重新载入模型参数,再同时在第一种仿真环境(图5(a))和第二种仿真环境(图5(b))中训练。

[0085] 本发明导航方法不需要复杂的前端感知模块,根据生成的栅格地图来决策,同时具有兼容不同传感器的优点。

[0086] 如图4所示,本发明技术在七种不同的仿真测试环境进行测试,同时使用传统的多机器人避障算法(NH-ORCA),以及其他研究人员最新提出的基于激光原始数据的避障算法(sensor-based)进行测试对比,本发明的方法在仅经过第一阶段训练后的模型(Map-based stage 1)以及在仅经过第二阶段训练后的模型(Map-based stage 2)进行对比,七种环境分别为:

[0087] (1)圆形环境(circle)(即圆形环境的仿真训练场景):机器人的起点匀称的初始化在一个圆上,每个机器人的目标点在圆的对面。本发明中使用4种圆形场景(参见图4(a)、图4(b)、图4(c)、图4(d)),机器人数量分别为6,8,10,12,圆形半径分别为2.5m,3m,3.5m,3.5m;

[0088] (2)交叉环境(cross)(即交叉环境的仿真训练场景):8个机器人分为两组,两组机器人的起点和终点连线之间呈十字形交叉(参见图4(e));

[0089] (3)交换环境(swap)(即交换环境的仿真训练场景):8机器人分为两组,每组机器人向另一组移动(参见图4(f));

[0090] (4)随机环境(random)(即随机环境的仿真训练场景):10个机器人的起点和终点在整个场景中随机选取(参见图4(g)、图4(h))。

[0091] 实验结果如下表所示,本发明技术在七种测试环境中多个指标优于相关方法,与现有其他方法相比,本发明的导航方法具有到达率高,所花费时间和平均路程短等优点。

[0092]

Scenarios (agents, range)	Method	$\bar{\pi}$	$\bar{t}$ (mean/std)	$\bar{d}$ (mean/std)	$\bar{v}$ (mean/std)
Circle (6, radius 2.5m)	NH-ORCA	0.969	2.6676/1.3981	**0.2004/0.1160**	0.4490/0.1537
	Sensor-level	**1.000**	2.0620/0.5576	0.8773/0.2269	0.5636/0.1328
	Map-based Stage 1	0.937	8.2528/6.4266	0.7861/0.4763	0.3328/0.2881
	Map-based Stage 2	**1.000**	**2.0000/0.3502**	0.8648/0.1447	**0.5659/0.1283**
Circle (8, radius 3m)	NH-ORCA	0.950	3.4988/1.9744	**0.2057/0.1299**	0.4479/0.1520
	Sensor-level	**1.000**	2.5400/0.5084	1.1992/0.1918	0.5687/0.1233
	Map-based Stage 1	0.914	10.3488/6.3236	0.9185/0.5446	0.3218/0.2880
	Map-based Stage 2	**1.000**	**2.3170/0.2577**	1.0204/0.1513	**0.5730/0.1146**
Circle (10, radius 3.5m)	NH-ORCA	0.892	4.2930/2.6132	**0.2486/0.1983**	0.4366/0.1546
	Sensor-level	**1.000**	3.3045/0.4784	1.5991/0.2145	0.5734/0.1142
	Map-based Stage 1	0.903	11.9304/9.2772	1.0635/0.6968	0.3212/0.2867
	Map-based Stage 2	**1.000**	**2.5881/0.4650**	1.1870/0.1710	**0.5735/0.1114**
Circle (12, radius 3.5m)	NH-ORCA	0.862	5.2137/3.4742	**0.2817/0.2599**	0.4078/0.1711
	Sensor-level	**1.000**	3.7290/0.5355	1.7884/0.2525	0.5699/0.1170
	Map-based Stage 1	0.873	15.7697/11.7475	1.0773/0.7475	0.2698/0.2871
	Map-based Stage 2	**1.000**	**2.6133/0.4527**	1.2170/0.1769	**0.5745/0.1120**
Cross ($8, 8 \times 8m^2$)	NH-ORCA	0.958	2.1283/1.5166	**0.1883/0.2081**	0.4851/0.1430
	Sensor-level	0.995	2.8238/1.2894	1.1174/0.5214	0.5419/0.1588
	Map-based Stage 1	0.950	4.0802/3.4952	1.0158/0.7322	0.4764/0.2278
	Map-based Stage 2	**1.000**	**1.8315/1.2333**	0.7873/0.4912	**0.5608/0.1384**
Swap ($8, 8 \times 8m^2$)	NH-ORCA	0.906	2.2174/2.1307	**0.2651/0.2228**	0.4845/0.1648
	Sensor-level	1.000	2.7357/0.9494	1.1498/0.3479	0.5535/0.1419
	Map-based Stage 1	0.994	2.7272/2.2479	0.8017/0.5761	0.5206/0.1874
	Map-based Stage 2	**1.000**	**2.0201/1.0430**	0.9816/0.3660	**0.5584/0.1424**
Random ($10, 8 \times 8m^2$)	NH-ORCA	0.934	4.3181/3.1353	0.5697/0.6412	0.3760/0.1890
	Sensor-level	0.924	3.4519/3.4162	0.5417/0.5048	0.4017/0.2687
	Map-based Stage 1	0.955	3.1650/2.5632	0.5514/0.4643	0.4202/0.2590
	Map-based Stage 2	**0.986**	**2.9009/2.4523**	**0.4531/0.3610**	**0.4460/0.2497**

[0093] 实施例

[0094] 本实施例提供一种分布式多机器人的导航方法，使用了四台机器人，均使用kobuki作为机器人底盘，nvidia Jetson TX2作为计算平台，分别使用Hokuyo UTM-30LX激光传感器和较为经济的Hokuyo URG-04LX激光传感器；Nvidia Jetson TX2安装了机器人操作系统ROS。提前使用了机器人操作系统ROS中Gmapping包生成了全局地图。

[0095] 在短距离导航避障中，只需使用粒子滤波器结合全局地图对机器人的位置进行估计，并直接向避障算法发送目标点即可。图5～6是机器人在短距离导航中的路径图。短距离导航场景说明如下：

[0096] 静态场景：场景里只有单个机器人以及静态的障碍物，但同时会有部分障碍物会突然出现在机器人面前；

[0097] 多机器人场景：场景中有多个机器人以及障碍物，每个机器人都需要避开其他机器人与场景中的障碍物到达各自的目标点；

[0098] 混合场景：场景中有机器人，行人。机器人在行驶的过程中不仅需要互相避开，也需要避开行人。

[0099] 为了应用到长距离导航中，使用粒子滤波器结合全局地图对机器人的位置进行估计，同时使用A星算法作为全局路径规划器生成整条全局路径并发送局部目标点。图7为长距离导航中的一个实例。机器人穿过整条走廊，期间安全地避开了静态障碍物以及行人。

[0100] 以上所述，仅为本发明较佳的具体实施方式，但本发明的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本发明披露的技术范围内，可轻易想到的变化或替换，都应涵盖在本发明的保护范围之内。因此，本发明的保护范围应该以权利要求书的保护范围为准。

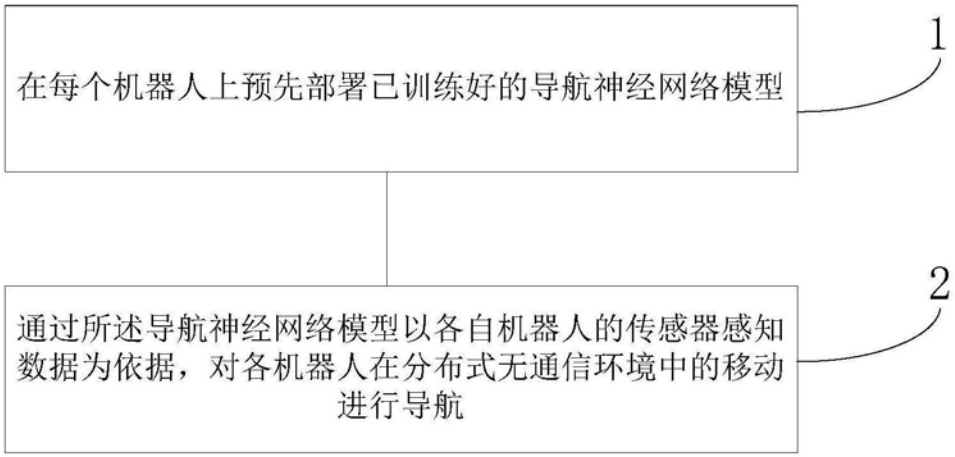

在每个机器人上预先部署已训练好的导航神经网络模型
1
通过所述导航神经网络模型以各自机器人的传感器感知数据为依据，对各机器人在分布式无通信环境中的移动进行导航
2

图1

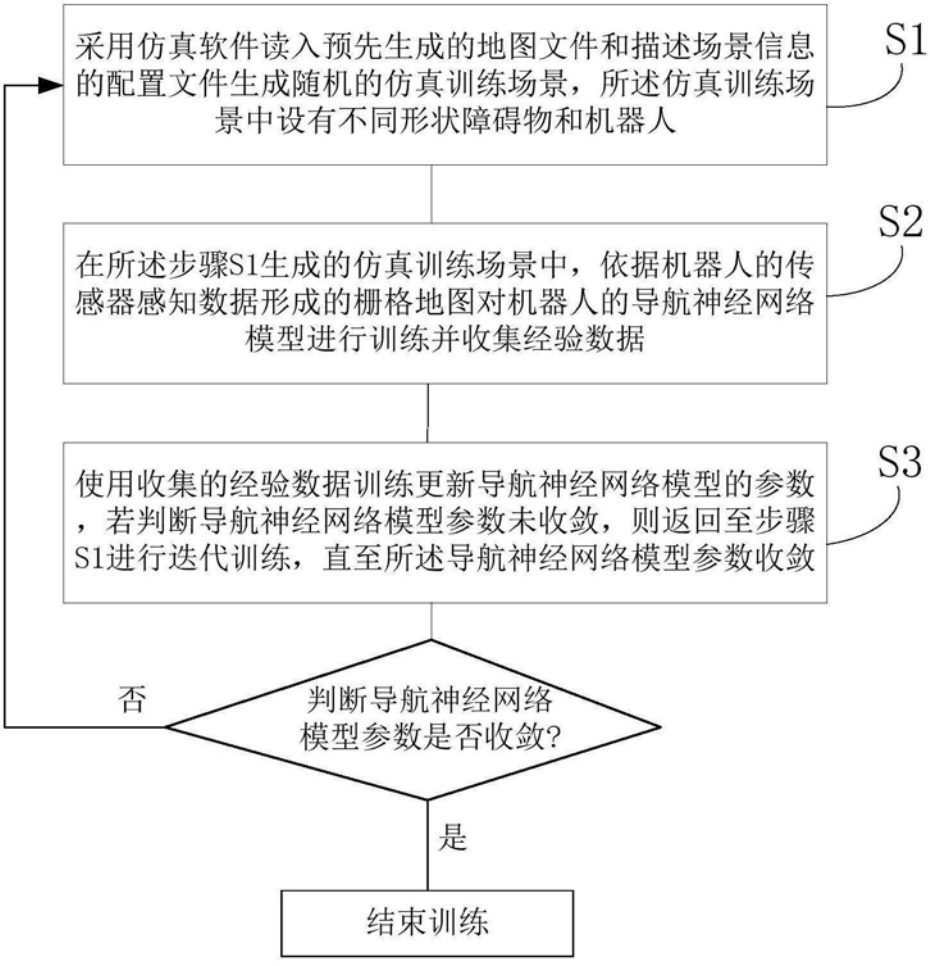

采用仿真软件读入预先生成的地图文件和描述场景信息的配置文件生成随机的仿真训练场景，所述仿真训练场景中设有不同形状障碍物和机器人
S1
在所述步骤S1生成的仿真训练场景中，依据机器人的传感器感知数据形成的栅格地图对机器人的导航神经网络模型进行训练并收集经验数据
S2
使用收集的经验数据训练更新导航神经网络模型的参数，若判断导航神经网络模型参数未收敛，则返回至步骤S1进行迭代训练，直至所述导航神经网络模型参数收敛
S3
否
判断导航神经网络模型参数是否收敛?
是
结束训练

图2

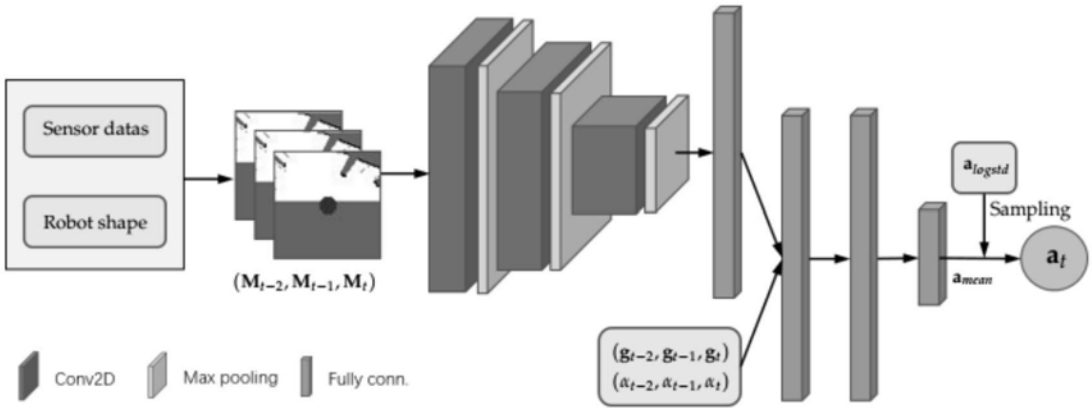

Sensor datas
Robot shape
$(\mathbf{M}_{t-2}, \mathbf{M}_{t-1}, \mathbf{M}_t)$
$(\mathbf{g}_{t-2}, \mathbf{g}_{t-1}, \mathbf{g}_t)$
$(\alpha_{t-2}, \alpha_{t-1}, \alpha_t)$
$\mathbf{a}_{logstd}$
Sampling
$\mathbf{a}_t$
$\mathbf{a}_{mean}$
Conv2D
Max pooling
Fully conn.

图3

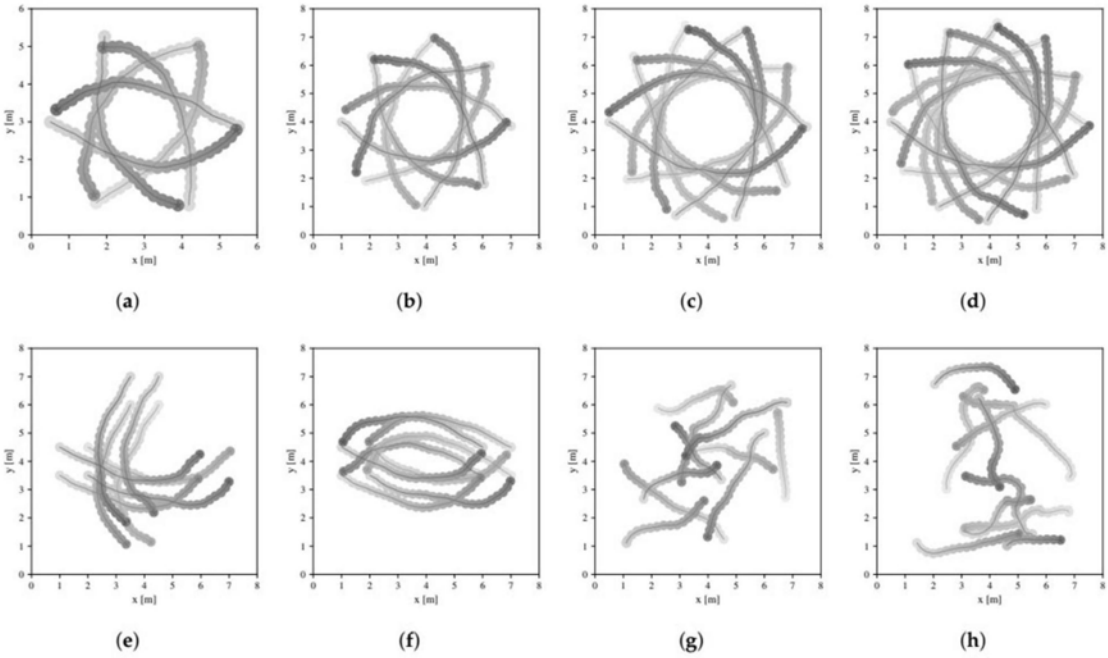

x [m]
y [m]
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)

图4

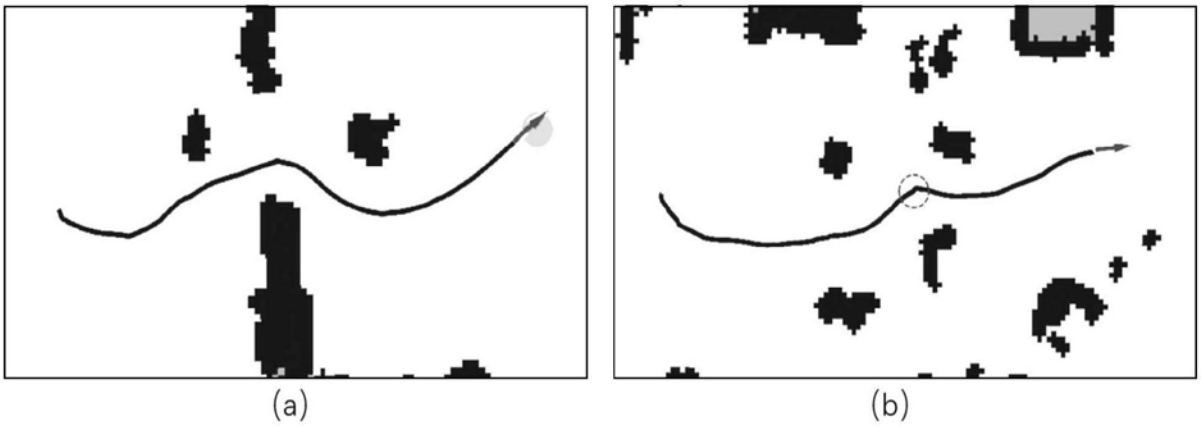


图5

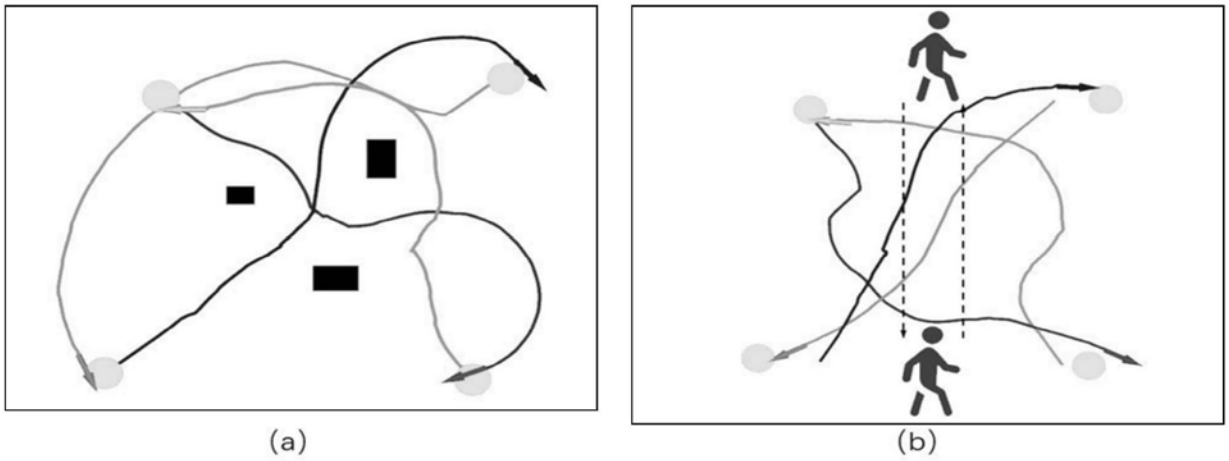


图6

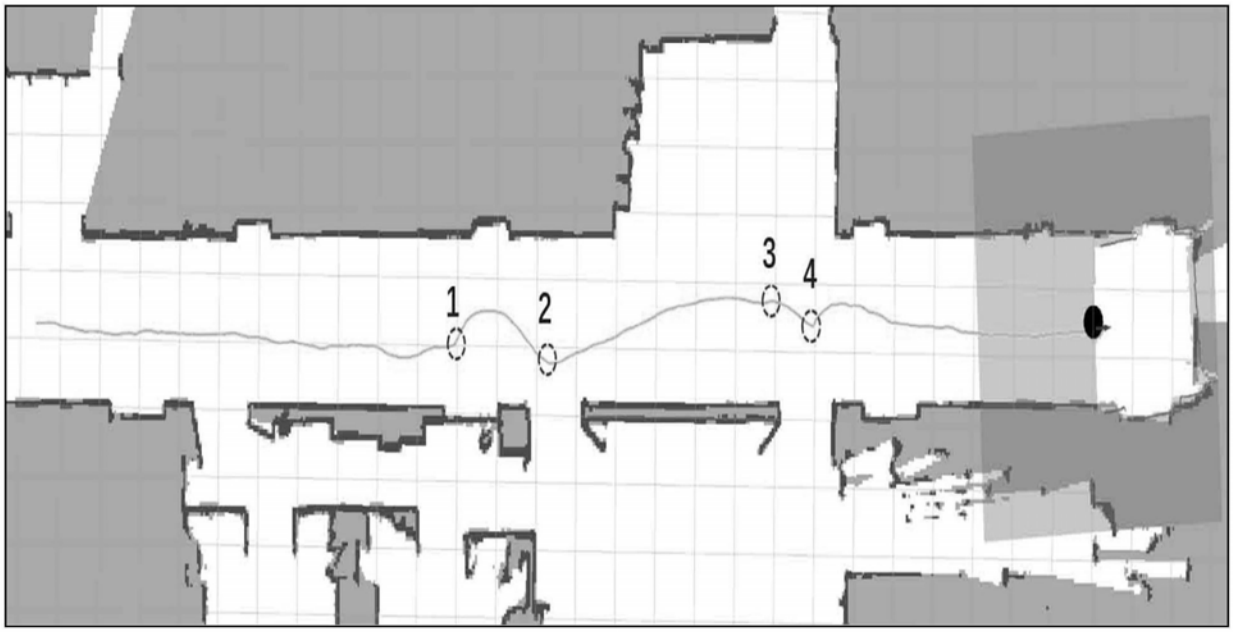


图7

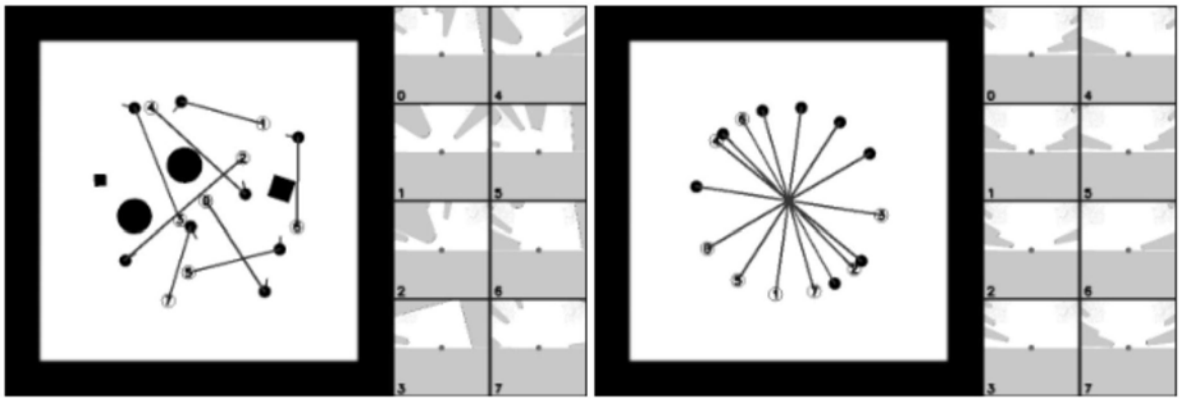


图8