



# (12) 发明专利

(10) 授权公告号 CN 114185354 B

(45) 授权公告日 2022.05.17

(21) 申请号 202210134890.X  
 (22) 申请日 2022.02.15  
 (65) 同一申请的已公布的文献号  
 申请公布号 CN 114185354 A  
 (43) 申请公布日 2022.03.15  
 (73) 专利权人 中国科学院微电子研究所  
 地址 100029 北京市朝阳区北土城西路3号  
 (72) 发明人 胡子骏 许绍云 汪洋  
 (74) 专利代理机构 北京天达知识产权代理事务  
 所(普通合伙) 11386  
 专利代理师 牛洪瑜  
 (51) Int.Cl.  
 G05D 1/02 (2020.01)

(56) 对比文件  
 CN 110883776 A, 2020.03.17  
 CN 110989570 A, 2020.04.10  
 CN 111310999 A, 2020.06.19  
 CN 113074738 A, 2021.07.06  
 US 2018089553 A1, 2018.03.29  
 US 2020174471 A1, 2020.06.04  
 审查员 李瞳玲

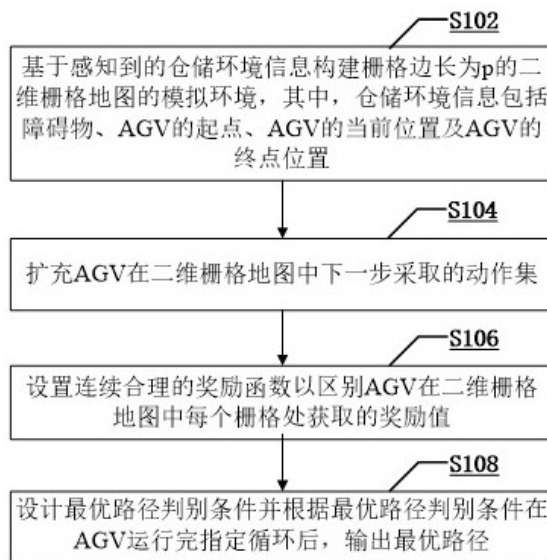
权利要求书4页 说明书12页 附图6页

## (54) 发明名称

一种基于DQN的AGV全局路径规划方法和系统

## (57) 摘要

本发明涉及一种基于DQN的AGV全局路径规划方法和系统,属于路径规划技术领域,解决了现有路径规划方法消耗时间长并且路径规划结构不准确的问题。该方法包括:基于感知到的仓储环境信息构建栅格边长为p的二维栅格地图的模拟环境,其中,仓储环境信息包括障碍物、AGV的起点、AGV的当前位置及AGV的终点位置;扩充AGV在二维栅格地图中下一步采取的动作集;设置连续合理的奖励函数以区别AGV在二维栅格地图中每个栅格处获取的奖励值;以及设计最优路径判别条件并根据最优路径判别条件在AGV运行完指定循环后,输出最优路径。结合增加的AGV运动方向,修改最优路径判别条件能够有效提升路径规划的准确率。通过设置连续合理的奖励函数能够提升路径规划的效率。



CN 114185354 B

1. 一种基于DQN的AGV全局路径规划方法,其特征在于,包括:

基于感知到的仓储环境信息构建栅格边长为 $p$ 的二维栅格地图的模拟环境,其中,所述仓储环境信息包括障碍物位置、所述AGV的起点、所述AGV的当前位置及所述AGV的终点位置;

扩充所述AGV在所述二维栅格地图中下一步采取的动作集;

设置连续合理的奖励函数以区别所述AGV在所述二维栅格地图中每个栅格处获取的奖励值,其中,通过以下公式设置连续合理的奖励函数reward:

$$\text{reward} = \begin{cases} -1, & \text{预计下一步遇到障碍物} \\ 1 - m * \left( \frac{\text{step distance}}{\text{long distance}} \right) + 1 - n * \text{corner}, & \text{预计下一步遇到目标点} \\ 1 - \frac{\text{next distance}}{\text{cur distance}}, & \text{其他} \end{cases}$$

其中, $\text{step distance}$ 表示预计本循环到达终点走过的总欧氏距离, $\text{long distance}$ 表示之前所有循环走到终点所走过的最长欧氏距离, $\text{corner}$ 表示本循环到达终点所进行的转弯数, $\text{cur distance}$ 表示这一步走完到达终点的欧氏距离, $\text{next distance}$ 表示下一步走完到达终点的欧氏距离, $m$ 、 $n$ 表示常数系数;以及

设计最优路径判别条件并根据所述最优路径判别条件在所述AGV运行完指定循环后,输出最优路径,其中,

所述最优路径保证在路径长度最短的前提下,转弯数最少:

$$d = \begin{cases} D, & D < d \\ d, & D > d \end{cases}$$

$$c = \begin{cases} \text{corner}_{\text{总}}, & \text{corner}_{\text{总}} < c \\ c, & \text{corner}_{\text{总}} > c \end{cases}$$

其中, $D$ 表示本循环AGV从起点到终点的总欧氏距离; $d$ 表示已知的AGV从起点到终点的最短欧氏距离; $\text{corner}_{\text{总}}$ 表示本循环AGV从起点到终点的转弯数; $c$ 表示已知的AGV从起点到终点的最少转弯数;最短欧氏距离 $d$ 和最少转弯数 $c$ 的初始设置值均为 $+\infty$ 。

2. 根据权利要求1所述的基于DQN的AGV全局路径规划方法,其特征在于,扩充所述AGV在所述二维栅格地图中下一步采取的动作集进一步包括:

将上、下、左、右4个方向的动作集扩充为上、下、左、右、左上、右上、左下、右下8个方向的动作集;以及

所述AGV通过执行扩充的动作集中的动作到达所述二维栅格地图中的8个临近栅格中的一个。

3. 根据权利要求2所述的基于DQN的AGV全局路径规划方法,其特征在于,所述AGV通过执行扩充的动作集中的动作到达所述二维栅格地图中的8个临近栅格中的一个,进一步包括:

所述AGV在当前位置坐标通过执行动作 $a$ 到达所述二维栅格地图中的邻近位置坐标:

$$(x_{i+1}, y_{i+1}) = \begin{cases} (x_i, y_i + p), & a = up \\ (x_i, y_i - p), & a = down \\ (x_i - p, y_i), & a = left \\ (x_i + p, y_i), & a = right \\ (x_i - p, y_i + p), & a = upleft \\ (x_i + p, y_i + p), & a = upright \\ (x_i - p, y_i - p), & a = downleft \\ (x_i + p, y_i - p), & a = downright \end{cases}$$

其中,  $(x_i, y_i)$  表示所述AGV当前位置坐标,  $(x_{i+1}, y_{i+1})$  表示所述AGV执行完动作集中动作 $a$ 后所处的邻近位置坐标;  $(x_i, y_i+p)$  和  $(x_i, y_i-p)$  分别表示在所述AGV当前位置坐标基础上,  $y$ 轴正和负方向添加一个栅格边长;  $(x_i+p, y_i)$  和  $(x_i-p, y_i)$  分别表示在所述AGV当前位置坐标基础上,  $x$ 轴正和负方向添加一个栅格边长。

4. 根据权利要求1所述的基于DQN的AGV全局路径规划方法, 其特征在于, 设置连续合理的奖励函数以区别所述AGV在所述二维栅格地图中每个栅格处获取的奖励值进一步包括:

根据以下Q-learning算法, 建立Q值表以记录所述AGV与所述模拟环境不断交互得到的奖励值;

通过迭代修改Q值表以增加选择正奖励值的概率, 使得所述AGV的每一步动作都趋向于所述扩充的动作集中的最优动作, 其中, 所述Q-learning算法的基本形式为:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t) \right]$$

其中,  $r_{t+1}$  代表状态 $s_t$ 情况下执行完动作 $a_t$ 后立刻得到的奖励,  $s_{t+1}$  表示 $s_t$ 执行完动作 $a_t$ 后的下一状态,  $a$  则代表状态 $s_{t+1}$ 情况下采取的所有动作,  $\gamma$  代表对未来奖励的衰减因子; 左侧的 $Q(s_t, a_t)$  代表修改后的Q值表, 右侧的 $Q(s_t, a_t)$  代表Q值表中已有的Q估计值,  $r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a)$  是Q现实值,  $\alpha$  代表学习率, 对所述Q估计值和所述Q现实值之间有多少误差进行学习;

DQN构建两个神经网络, 通过神经网络来代替Q值表, 一个用于计算Q估计值, 另一个用于计算Q现实值。每个网络都由输入层、隐藏层和输出层构成, 所述输入层输入所述AGV当前位置坐标, 所述隐藏层含有 $n$ 个神经元, 以及所述输出层输出当前位置坐标的所有Q值, 根据 $\epsilon$ -greedy原则做出决策, 得到执行所述扩充的动作集中的动作 $a$ 的邻近位置坐标作为AGV下一步的环境坐标。

5. 根据权利要求1所述的基于DQN的AGV全局路径规划方法, 其特征在于, 所述AGV每做出一个动作会获得一个奖励值, 所述AGV在所述扩充的动作集中选取合理动作以获得更高奖励值。

6. 根据权利要求1所述的基于DQN的AGV全局路径规划方法, 其特征在于, 根据所述最优路径判别条件在所述AGV运行完指定循环后, 输出最优路径进一步包括:

通过判断所述AGV在所述二维栅格地图中实际行驶的欧氏距离及最终路径转弯数,确定本循环路径中的最优路径。

7. 根据权利要求6所述的基于DQN的AGV全局路径规划方法,其特征在于,通过判断所述AGV在所述二维栅格地图中实际行驶的欧氏距离及最终路径转弯数进一步包括:对所述AGV上一步、当前、下一步共三个栅格的坐标进行计算,判断所述AGV进行直线运动还是进行转弯运动,其中,当所述AGV满足沿水平直线方向运动、沿竖直直线方向运动和沿对角线方向运动中的一种时,所述AGV走直线运动:

所述沿水平直线方向运动:

$$(x_{i-1} - p, y_{i-1}) = (x_i, y_i) = (x_{i+1} + p, y_{i+1})$$

$$(x_{i-1} + p, y_{i-1}) = (x_i, y_i) = (x_{i+1} - p, y_{i+1})$$

所述沿竖直直线方向运动:

$$(x_{i-1}, y_{i-1} - p) = (x_i, y_i) = (x_{i+1}, y_{i+1} + p)$$

$$(x_{i-1}, y_{i-1} + p) = (x_i, y_i) = (x_{i+1}, y_{i+1} - p)$$

所述沿对角线方向运动:

$$(x_{i-1} - p, y_{i-1} - p) = (x_i, y_i) = (x_{i+1} + p, y_{i+1} + p)$$

$$(x_{i-1} - p, y_{i-1} + p) = (x_i, y_i) = (x_{i+1} + p, y_{i+1} - p)$$

$$(x_{i-1} + p, y_{i-1} + p) = (x_i, y_i) = (x_{i+1} - p, y_{i+1} - p)$$

$$(x_{i-1} + p, y_{i-1} - p) = (x_i, y_i) = (x_{i+1} - p, y_{i+1} + p)$$

其中,  $(x_i, y_i)$  表示AGV当前所在位置坐标,  $(x_{i+1}, y_{i+1})$  表示AGV执行完动作后所处的位置坐标;  $(x_{i-1}, y_{i-1})$  表示AGV上一步所在位置坐标;  $(x_i, y_i + p)$  表示在AGV当前所在位置坐标基础上,  $y$ 轴方向添加一个栅格边长;  $(x_i + p, y_i)$  表示在AGV当前所在位置坐标基础上,  $x$ 轴方向添加一个栅格边长。

8. 根据权利要求7所述的基于DQN的AGV全局路径规划方法,其特征在于,当所述AGV满足沿水平直线方向运动、沿竖直直线方向运动和沿对角线方向运动中的两种时,所述AGV出现转弯并且从起点开始的整个路径转弯数加一:

$$corner = corner + 1$$

其中,  $corner$  代表所述本循环路径中已有的转弯数;

所述AGV从起点到终点在所述二维栅格地图中的实际欧氏距离:

$$D = \sum_{j=1}^u \sqrt{(y_j - y_{j-1})^2 + (x_j - x_{j-1})^2}$$

其中, $D$ 表示所述AGV本循环从起点到终点在二维栅格地图中行驶的总欧氏距离; $u$ 表示所述AGV行驶的总栅格数,包含起点和终点一共经过 $u+1$ 个栅格; $(x_j, y_j)$ 表示所述AGV经过的第 $j$ 个栅格坐标。

9.一种基于DQN的AGV全局路径规划系统,其特征在于,包括:

栅格地图构建模块,用于基于感知到的仓储环境信息构建栅格边长为 $p$ 的二维栅格地图的模拟环境,其中,所述仓储环境信息包括障碍物、所述AGV的起点、所述AGV的当前位置及所述AGV的终点位置;

动作集扩充模块,用于扩充所述AGV在所述二维栅格地图中下一步采取的动作集;

奖励函数设置模块,用于设置连续合理的奖励函数以区别所述AGV在所述二维栅格地图中每个栅格处获取的奖励值,其中,通过以下公式设置连续合理的奖励函数reward:

$$\text{reward} = \begin{cases} -1, & \text{预计下一步遇到障碍物} \\ 1 - m * \left( \frac{\text{step distance}}{\text{long distance}} \right) + 1 - n * \text{corner}, & \text{预计下一步遇到目标点} \\ 1 - \frac{\text{next distance}}{\text{cur distance}}, & \text{其他} \end{cases}$$

其中, $\text{step distance}$ 表示预计本循环到达终点走过的总欧氏距离, $\text{long distance}$ 表示之前所有循环走到终点所走过的最长欧氏距离, $\text{corner}$ 表示本循环到达终点所进行的转弯数, $\text{cur distance}$ 表示这一步走完到达终点的欧氏距离, $\text{next distance}$ 表示下一步走完到达终点的欧氏距离, $m$ 、 $n$ 表示常数系数;以及

最优路径输出模块,用于设计最优路径判别条件并根据所述最优路径判别条件在所述AGV运行完指定循环后,输出最优路径,其中,所述最优路径保证在路径长度最短的前提下,转弯数最少:

$$d = \begin{cases} D, & D < d \\ d, & D > d \end{cases}$$

$$c = \begin{cases} \text{corner}_{\text{总}}, & \text{corner}_{\text{总}} < c \\ c, & \text{corner}_{\text{总}} > c \end{cases}$$

其中, $D$ 表示本循环AGV从起点到终点的总欧氏距离; $d$ 表示已知的AGV从起点到终点的最短欧氏距离; $\text{corner}_{\text{总}}$ 表示本循环AGV从起点到终点的转弯数; $c$ 表示已知的AGV从起点到终点的最少转弯数;最短欧氏距离 $d$ 和最少转弯数 $c$ 的初始设置值均为 $+\infty$ 。

## 一种基于DQN的AGV全局路径规划方法和系统

### 技术领域

[0001] 本发明涉及路径规划技术领域,尤其涉及一种基于DQN的AGV全局路径规划方法和系统。

### 背景技术

[0002] 随着智能仓储技术的发展,仓库中货物的起重、运输、卸包、装包等日常工作都可以交给AGV(移动机器人)完成,可提升仓库中货品的存取效率。合理的路径规划可以提升AGV的工作效率,降低AGV的故障发生率。现有的路径规划算法主要包括:基于传统图论的Dijkstra算法、A\*算法;蚁群、粒子群、遗传等仿生学算法。上述算法存在易陷入局部最优等缺陷,难以应对复杂的仓储环境。

[0003] 在已知仓储环境中,利用DQN(即,深度Q网络)进行全局路径规划是近年来出现的新方法,可以有效避免出现局部最优情况。在仓储环境的栅格图下,采用传统DQN进行路径规划,会出现路径规划过程消耗时间长、路径规划结果不准确的问题。其中,时间长是因为AGV行驶得到的奖励反馈不明显,需要进行更多的“试错”;结果不准确是因为给予AGV选择方向不多,且路径判别条件不够准确。

### 发明内容

[0004] 鉴于上述的分析,本发明实施例旨在提供一种基于DQN的AGV全局路径规划方法和系统,用以解决现有路径规划方法消耗时间长并且路径规划结构不准确的问题。

[0005] 一方面,本发明实施例提供了一种基于DQN的AGV全局路径规划方法,包括:基于感知到的仓储环境信息构建栅格边长为 $p$ 的二维栅格地图的模拟环境,其中,所述仓储环境信息包括障碍物位置、所述AGV的起点、所述AGV的当前位置及所述AGV的终点位置;扩充所述AGV在所述二维栅格地图中下一步采取的动作集;设置连续合理的奖励函数以区别所述AGV在所述二维栅格地图中每个栅格处获取的奖励值;以及设计最优路径判别条件并根据所述最优路径判别条件在所述AGV运行完指定循环后,输出最优路径。

[0006] 上述技术方案的有益效果如下:本申请通过将传统DQN在栅格图中下一步能采取的动作由4个扩充为8个,能够增加路径规划的准确性,扩充AGV动作集。通过设置连续合理的奖励函数reward,能够保证AGV在每一个栅格获取奖励值的区别,减少AGV无效探索的时间,能够缩短基于DQN的AGV全局路径规划时间,提升路径规划的效率。结合增加的AGV运动方向,修改最优路径判别条件能够有效提升路径规划的准确率。

[0007] 基于上述方法的进一步改进,扩充所述AGV在所述二维栅格地图中下一步采取的动作集进一步包括:将上、下、左、右4个方向的动作集扩充为上、下、左、右、左上、右上、左下、右下8个方向的动作集;以及所述AGV通过执行扩充的动作集中的动作到达所述二维栅格地图中的8个临近栅格中的一个。

[0008] 基于上述方法的进一步改进,所述AGV通过执行扩充的动作集中的动作到达所述二维栅格地图中的8个临近栅格中的一个进一步包括:所述AGV在所述当前位置坐标通过执

行动作 $a$ 到达所述二维栅格地图中的邻近位置坐标：

$$[0009] \quad (x_{i+1}, y_{i+1}) = \begin{cases} (x_i, y_i + p), & a = up \\ (x_i, y_i - p), & a = down \\ (x_i - p, y_i), & a = left \\ (x_i + p, y_i), & a = right \\ (x_i - p, y_i + p), & a = upleft \\ (x_i + p, y_i + p), & a = upright \\ (x_i - p, y_i - p), & a = downleft \\ (x_i + p, y_i - p), & a = downright \end{cases}$$

[0010] 其中,  $(x_i, y_i)$  表示所述AGV当前位置坐标,  $(x_{i+1}, y_{i+1})$  表示所述AGV执行完动作集中动作 $a$ 后所处的邻近位置坐标;  $(x_i, y_i+p)$  和  $(x_i, y_i-p)$  分别表示在所述AGV当前所在位置坐标基础上,  $y$ 轴正和负方向添加一个栅格边长;  $(x_i+p, y_i)$  和  $(x_i-p, y_i)$  分别表示在所述AGV当前所在位置坐标基础上,  $x$ 轴正和负方向添加一个栅格边长。

[0011] 基于上述方法的进一步改进, 设置连续合理的奖励函数以区别所述AGV在所述二维栅格地图中每个栅格处获取的奖励值进一步包括: 根据以下Q-learning算法, 建立Q值表以记录所述AGV与所述模拟环境不断交互得到的奖励值; 通过迭代修改Q值表以增加选择正奖励值的概率, 使得所述AGV的每一步动作都趋向于所述扩充的动作集中的最优动作, 其中, 所述Q-learning算法的基本形式为:

$$[0012] \quad Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t) \right]$$

[0013] 其中,  $r_{t+1}$  代表状态 $s_t$ 情况下执行完动作 $a_t$ 后立刻得到的奖励,  $s_{t+1}$  表示 $s_t$ 执行完动作 $a_t$ 后的下一状态,  $a$  则代表状态 $s_{t+1}$ 情况下采取的所有动作,  $\gamma$  代表对未来奖励的衰减; 左侧的 $Q(s_t, a_t)$  代表修改后的Q值表, 右侧的 $Q(s_t, a_t)$  代表Q值表中已有的Q估计值,  $r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a)$  是Q现实值,  $\alpha$  代表学习率, 对所述Q估计值和所述Q现实值之间有多少误差进行学习; DQN则构建两个神经网络, 通过神经网络来代替Q值表, 一个用于计算Q估计值, 另一个用于计算Q现实值。每个网络都由输入层、隐藏层和输出层构成, 所述输入层输入所述AGV当前位置坐标, 所述隐藏层含有 $n$ 个神经元, 以及所述输出层输出当前坐标的所有Q值, 根据 $\epsilon$ -greedy原则做出决策, 得到执行所述扩充的动作集中的动作 $a$ 的邻近位置坐标作为AGV下一步的环境坐标。

[0014] 基于上述方法的进一步改进, 通过以下公式设置连续合理的奖励函数reward:

$$[0015] \quad reward = \begin{cases} -1, & \text{预计下一步遇到障碍物} \\ 1 - m * \left( \frac{step\ distance}{long\ distance} \right) + 1 - n * corner, & \text{预计下一步遇到目标点} \\ 1 - \frac{next\ distance}{cur\ distance}, & \text{其他} \end{cases}$$

[0016] 其中, *step distance* 表示预计本循环到达终点走过的总欧氏距离, *long distance* 表示之前所有循环走到终点所走过的最长欧氏距离, *corner* 表示本循环到达终点所进行的转弯数, *cur distance* 表示这一步走完到达终点的欧氏距离, *next distance* 表示下一步走完到达终点的欧氏距离, *m*、*n* 表示常数系数。

[0017] 基于上述方法的进一步改进, 所述AGV每做出一个动作会获得一个奖励值, 所述AGV在所述扩充的动作集中选取合理动作以获得更高奖励值。

[0018] 基于上述方法的进一步改进, 根据所述最优路径判别条件在所述AGV运行完指定循环后, 输出最优路径进一步包括: 通过判断所述AGV在所述二维栅格地图中实际行驶的欧氏距离及最终路径转弯数, 确定本循环路径中的最优路径。

[0019] 基于上述方法的进一步改进, 通过判断所述AGV在所述二维栅格地图中实际行驶的欧氏距离及最终路径转弯数进一步包括: 对所述AGV上一步、当前、下一步共三个栅格的坐标进行计算, 判断所述AGV进行直线运动还是进行转弯运动, 其中, 当所述AGV满足沿水平直线方向运动、沿竖直直线方向运动和沿对角线方向运动中的一种时, 所述AGV走直线运动: 所述沿水平直线方向运动:

$$\begin{aligned} [0020] \quad & (x_{i-1} - p, y_{i-1}) = (x_i, y_i) = (x_{i+1} + p, y_{i+1}) \\ & (x_{i-1} + p, y_{i-1}) = (x_i, y_i) = (x_{i+1} - p, y_{i+1}) \end{aligned}$$

[0021] 所述沿竖直直线方向运动:

$$\begin{aligned} [0022] \quad & (x_{i-1}, y_{i-1} - p) = (x_i, y_i) = (x_{i+1}, y_{i+1} + p) \\ & (x_{i-1}, y_{i-1} + p) = (x_i, y_i) = (x_{i+1}, y_{i+1} - p) \end{aligned}$$

[0023] 所述沿对角线方向运动:

$$\begin{aligned} [0024] \quad & (x_{i-1} - p, y_{i-1} - p) = (x_i, y_i) = (x_{i+1} + p, y_{i+1} + p) \\ & (x_{i-1} - p, y_{i-1} + p) = (x_i, y_i) = (x_{i+1} + p, y_{i+1} - p) \\ & (x_{i-1} + p, y_{i-1} + p) = (x_i, y_i) = (x_{i+1} - p, y_{i+1} - p) \\ & (x_{i-1} + p, y_{i-1} - p) = (x_i, y_i) = (x_{i+1} - p, y_{i+1} + p) \end{aligned}$$

[0025] 其中,  $(x_i, y_i)$  表示AGV当前所在位置坐标,  $(x_{i+1}, y_{i+1})$  表示AGV执行完动作后所处的位置坐标;  $(x_{i-1}, y_{i-1})$  表示AGV上一步所在位置坐标;  $(x_i, y_i + p)$  表示在AGV当前所在位置坐标基础上, *y*轴方向添加一个栅格边长;  $(x_i + p, y_i)$  表示在AGV当前所在位置坐标基础上, *x*轴方向添加一个栅格边长。

[0026] 基于上述方法的进一步改进, 当所述AGV满足沿水平直线方向运动、沿竖直直线方向运动和沿对角线方向运动中的两种时, 所述AGV出现转弯并且从起点开始的整个路径转弯数加一:



[0027]  $corner = corner + 1,$

[0028] 其中,  $corner$  代表所述本循环路径中已有的转弯数; 所述AGV从起点到终点在所述二维栅格地图中的实际欧氏距离:

$$[0029] \quad D = \sum_{j=1}^u \sqrt{(y_j - y_{j-1})^2 + (x_j - x_{j-1})^2}$$

[0030] 其中,  $D$  表示所述AGV本循环从起点到终点在二维栅格地图中行驶的总欧氏距离;  $u$  表示所述AGV行驶的总栅格数, 包含起点和终点一共经过  $u+1$  个栅格;  $(x_j, y_j)$  表示所述AGV经过的第  $j$  个栅格坐标; 所述AGV最优路径保证在路径长度最短的前提下, 转弯数最少:

$$[0031] \quad d = \begin{cases} D, & D < d \\ d, & D > d \end{cases}$$

$$[0032] \quad c = \begin{cases} corner_{总}, & corner_{总} < c \\ c, & corner_{总} > c \end{cases}$$

[0033] 其中,  $D$  表示本循环AGV从起点到终点的总欧氏距离;  $d$  表示已知的AGV从起点到终点的最短欧氏距离;  $corner_{总}$  表示本循环AGV从起点到终点的转弯数;  $c$  表示已知的AGV从起点到终点的最少转弯数; 最短欧氏距离  $d$  和最少转弯数  $c$  的初始设置值均为  $+\infty$ 。

[0034] 另一方面, 本发明实施例提供了一种基于DQN的AGV全局路径规划系统包括: 栅格地图构建模块, 用于基于感知到的仓储环境信息构建栅格边长为  $p$  的二维栅格地图的模拟环境, 其中, 所述仓储环境信息包括障碍物、所述AGV的起点、所述AGV的当前位置及所述AGV的终点位置; 动作集扩充模块, 用于扩充所述AGV在所述二维栅格地图中下一步采取的动作集; 奖励函数设置模块, 用于设置连续合理的奖励函数以区别所述AGV在所述二维栅格地图中每个栅格处获取的奖励值; 以及最优路径输出模块, 用于设计最优路径判别条件并根据所述最优路径判别条件在所述AGV运行完指定循环后, 输出最优路径。

[0035] 与现有技术相比, 本发明至少可实现如下有益效果之一:

[0036] 1、通过将传统DQN在栅格图中下一步能采取的动作由4个扩充为8个, 能够增加路径规划的准确性, 扩充AGV动作集。结合增加的AGV运动方向, 修改最优路径判别条件能够有效提升路径规划的准确率;

[0037] 2、通过设置连续合理的奖励函数  $reward$ , 能够保证AGV在每一个栅格获取奖励值的区别, 减少AGV无效探索的时间, 能够缩短基于DQN的AGV全局路径规划时间, 提升路径规划的效率;

[0038] 3、在DQN输出8个方向的前提下, 通过判断AGV在栅格图中实际行驶的欧氏距离及最终路径转弯数, 确定本回合内所能得到的最优路径, 能够增加路径规划的准确性。

[0039] 本发明中, 上述各技术方案之间还可以相互组合, 以实现更多的优选组合方案。本发明的其他特征和优点将在随后的说明书中阐述, 并且, 部分优点可从说明书中变得显而易见, 或者通过实施本发明而了解。本发明的目的和其他优点可通过说明书以及附图中所特别指出的内容中来实现和获得。

## 附图说明

- [0040] 附图仅用于示出具体实施例的目的,而并不认为是对本发明的限制,在整个附图中,相同的参考符号表示相同的部件。
- [0041] 图1为根据本发明实施例的基于DQN的AGV全局路径规划方法的流程图;
- [0042] 图2为根据本发明实施例的基于DQN的AGV全局路径规划方法的具体流程图;
- [0043] 图3为根据本发明实施例的全局路径规划中使用的DQN网络简易模型;
- [0044] 图4a和图4b分别为传统栅格图中AGV行驶方向的4个候选角度的示意图和根据本发明实施例的栅格图中AGV行驶方向的8个候选角度的示意图;
- [0045] 图5a和图5b分别为传统AGV网格行驶图中的沿栅格线方向经过3个栅格的示意图和根据本发明实施例的AGV网格行驶图中的沿对角线方向经过3个栅格的示意图;
- [0046] 图6为根据本发明实施例的仓储环境的栅格图;
- [0047] 图7a、图7b和图7c分别为根据本发明实施例的不包含转弯判别条件的三种AGV最优路径示意图;
- [0048] 图8a和图8b分别为根据本发明实施例的基于DQN的全局路径规划方法的两种栅格仿真结果的示意图;
- [0049] 图9为根据本发明实施例的路径规划方法的流程示意图;
- [0050] 图10为根据本发明实施例的基于DQN的AGV全局路径规划系统的框图。

## 具体实施方式

- [0051] 下面结合附图来具体描述本发明的优选实施例,其中,附图构成本申请一部分,并与本发明的实施例一起用于阐释本发明的原理,并非用于限定本发明的范围。
- [0052] 本发明的一个具体实施例,公开了一种基于DQN的AGV全局路径规划方法。参考图1,基于DQN的AGV全局路径规划方法包括:在步骤S102中,基于感知到的仓储环境信息构建栅格边长为 $p$ 的二维栅格地图的模拟环境,其中,仓储环境信息包括障碍物位置、AGV的起点、AGV的当前位置及AGV的终点位置, $p$ 为二维栅格地图中的单位栅格的边长;在步骤S104中,扩充AGV在二维栅格地图中下一步采取的动作集;在步骤S106中,设置连续合理的奖励函数以区别AGV在二维栅格地图中每个栅格处获取的奖励值;以及在步骤S108中,设计最优路径判别条件并根据最优路径判别条件在AGV运行完指定循环后,输出最优路径。
- [0053] 与现有技术相比,本实施例提供的基于DQN的AGV全局路径规划方法中,通过将传统DQN在栅格图中下一步能采取的动作进行扩充,能够增加路径规划的准确性,扩充AGV动作集。通过设置连续合理的奖励函数reward,能够保证AGV在每一个栅格获取奖励值的区别,减少AGV无效探索的时间,能够缩短基于DQN的AGV全局路径规划时间,提升路径规划的效率。结合增加的AGV运动方向,修改最优路径判别条件能够有效提升路径规划的准确率。
- [0054] 下文中,将参考图1对基于DQN的AGV全局路径规划方法的各个步骤进行详细描述。
- [0055] 在步骤S102中,基于感知到的仓储环境信息构建栅格边长为 $p$ 的二维栅格地图的模拟环境,其中,仓储环境信息包括障碍物位置、AGV的起点、AGV的当前位置及AGV的终点位置。具体地, $p$ 为二维栅格地图中的单位栅格的边长。
- [0056] 在步骤S104中,扩充AGV在二维栅格地图中下一步采取的动作集。扩充AGV在二维栅格地图中下一步采取的动作集进一步包括:将上、下、左、右4个方向的动作集扩充为上、

下、左、右、左上、右上、左下、右下8个方向的动作集;以及AGV通过执行扩充的动作集中的动作到达二维栅格地图中的8个临近栅格中的一个。

[0057] 具体地,AGV通过执行扩充的动作集中的动作到达二维栅格地图中的8个临近栅格中的一个,进一步包括:

[0058] AGV在当前位置坐标通过执行动作*a*到达二维栅格地图中的邻近位置坐标:

$$[0059] \quad (x_{i+1}, y_{i+1}) = \begin{cases} (x_i, y_i + p), & a = up \\ (x_i, y_i - p), & a = down \\ (x_i - p, y_i), & a = left \\ (x_i + p, y_i), & a = right \\ (x_i - p, y_i + p), & a = upleft \\ (x_i + p, y_i + p), & a = upright \\ (x_i - p, y_i - p), & a = downleft \\ (x_i + p, y_i - p), & a = downright \end{cases}$$

[0060] 其中,  $(x_i, y_i)$  表示所述AGV当前位置坐标,  $(x_{i+1}, y_{i+1})$  表示所述AGV执行完动作集中动作*a*后所处的邻近位置坐标;*a*代表动作集中的所有动作;  $(x_i, y_i + p)$  和  $(x_i, y_i - p)$  分别表示在所述AGV当前所在位置坐标基础上,*y*轴正和负方向添加一个栅格边长;  $(x_i + p, y_i)$  和  $(x_i - p, y_i)$  分别表示在所述AGV当前所在位置坐标基础上,*x*轴正和负方向添加一个栅格边长。

[0061] 在步骤S106中,设置连续合理的奖励函数以区别AGV在二维栅格地图中每个栅格处获取的奖励值。设置连续合理的奖励函数以区别AGV在二维栅格地图中每个栅格处获取的奖励值进一步包括:根据以下Q-learning算法,建立Q值表以记录AGV与模拟环境不断交互得到的奖励值;通过迭代修改Q值表以增加选择正奖励值的概率,使得AGV的每一步动作都趋向于扩充的动作集中的最优动作,其中,Q-learning算法的基本形式为:

$$[0062] \quad Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t) \right]$$

[0063] 其中,  $r_{t+1}$  代表状态  $s_t$  情况下执行完动作  $a_t$  后立刻得到的奖励,  $s_{t+1}$  表示  $s_t$  执行完动作  $a_t$  后的下一状态,  $a$  则代表状态  $s_{t+1}$  情况下采取的所有动作,  $\gamma$  代表对未来奖励的衰减因子;左侧的  $Q(s_t, a_t)$  代表修改后的Q值表,右侧的  $Q(s_t, a_t)$  代表Q值表中已有的Q估计值,  $r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a)$  是Q现实值,  $\alpha$  代表学习率,对Q估计值和Q现实值之间有多少

误差进行学习;构建两个深度Q网络DQN,通过DQN代替Q值表,一个DQN用于计算Q估计值,另一个DQN用于计算Q现实值,其中,每个DQN由输入层、隐藏层和输出层构成,输入层输入AGV当前位置坐标,隐藏层含有*n*个神经元,以及输出层输出执行扩充的动作集中的动作*a*的邻近位置坐标。具体地,DQN自身包括两个神经网络(两个Q网络),都能输出Q值,但有一个网络可以被称为target Q网络,其与另一个Q网络结构一样,初始的权重也一样,只是Q网络每次迭代都会更新,而target Q网络是每隔一段时间才会更新,权重更新较慢,因此权重记为

$\omega^-$ 。target Q网络计算Q值用于产生误差用于学习、辅助网络权重的更新，Q网络输出的值决定Q值，以及下一步AGV坐标。通过以下公式设置连续合理的奖励函数reward：

$$[0064] \quad reward = \begin{cases} -1, & \text{预计下一步遇到障碍物} \\ 1 - m * \left( \frac{step \ distance}{long \ distance} \right) + 1 - n * corner, & \text{预计下一步遇到目标点} \\ 1 - \frac{next \ distance}{cur \ distance}, & \text{其他} \end{cases}$$

[0065] 其中，*step distance*表示预计本循环到达终点走过的总欧氏距离，*long distance*表示之前所有循环走到终点所走过的最长欧氏距离，*corner*表示本循环到达终点所进行的转弯数，*cur distance*表示这一步走完到达终点的欧氏距离，*next distance*表示下一步走完到达终点的欧氏距离，*m*、*n*表示常数系数，在实例中取*m*=0.9，*n*=0.01。AGV每做出一个动作会获得一个奖励值，AGV在扩充的动作集中选取合理动作以获得更高奖励值。

[0066] 在步骤S108中，设计最优路径判别条件并根据最优路径判别条件在AGV运行完指定循环后，输出最优路径。根据最优路径判别条件在AGV运行完指定循环后，输出最优路径进一步包括：通过判断AGV在二维栅格地图中实际行驶的欧氏距离及最终路径转弯数，确定本循环路径中的最优路径。

[0067] 具体地，通过判断AGV在二维栅格地图中实际行驶的欧氏距离及最终路径转弯数进一步包括：对AGV上一步、当前、下一步共三个栅格的坐标进行计算，判断AGV进行直线运动还是进行转弯运动，其中，当AGV满足沿水平直线方向运动、沿竖直直线方向运动和沿对角线方向运动中的一种时，AGV走直线运动：

[0068] 沿水平直线方向运动：

$$[0069] \quad \begin{aligned} (x_{i-1} - p, y_{i-1}) &= (x_i, y_i) = (x_{i+1} + p, y_{i+1}) \\ (x_{i-1} + p, y_{i-1}) &= (x_i, y_i) = (x_{i+1} - p, y_{i+1}) \end{aligned}$$

[0070] 沿竖直直线方向运动：

$$[0071] \quad \begin{aligned} (x_{i-1}, y_{i-1} - p) &= (x_i, y_i) = (x_{i+1}, y_{i+1} + p) \\ (x_{i-1}, y_{i-1} + p) &= (x_i, y_i) = (x_{i+1}, y_{i+1} - p) \end{aligned}$$

[0072] 沿对角线方向运动：

$$[0073] \quad \begin{aligned} (x_{i-1} - p, y_{i-1} - p) &= (x_i, y_i) = (x_{i+1} + p, y_{i+1} + p) \\ (x_{i-1} - p, y_{i-1} + p) &= (x_i, y_i) = (x_{i+1} + p, y_{i+1} - p) \\ (x_{i-1} + p, y_{i-1} + p) &= (x_i, y_i) = (x_{i+1} - p, y_{i+1} - p) \\ (x_{i-1} + p, y_{i-1} - p) &= (x_i, y_i) = (x_{i+1} - p, y_{i+1} + p) \end{aligned}$$

[0074] 其中,  $(x_i, y_i)$  表示AGV当前所在位置坐标,  $(x_{i+1}, y_{i+1})$  表示AGV执行完动作后所处的位置坐标;  $(x_{i-1}, y_{i-1})$  表示AGV上一步所在位置坐标;  $(x_i, y_i+p)$  和  $(x_i, y_i-p)$  分别表示在AGV当前所在位置坐标基础上,  $y$ 轴正和负方向添加一个栅格边长;  $(x_i+p, y_i)$  和  $(x_i-p, y_i)$  分别表示在AGV当前所在位置坐标基础上,  $x$ 轴正和负方向添加一个栅格边长。

[0075] 当AGV满足沿水平直线方向运动、沿竖直直线方向运动和沿对角线方向运动中的两种时, AGV出现转弯并且从起点开始的整个路径转弯数加一:

$$[0076] \quad \mathit{corner} = \mathit{corner} + 1$$

[0077] 其中,  $\mathit{corner}$ 代表本循环路径中已有的转弯数;

[0078] AGV从起点到终点在二维栅格地图中的实际欧氏距离:

$$[0079] \quad D = \sum_{j=1}^u \sqrt{(y_j - y_{j-1})^2 + (x_j - x_{j-1})^2}$$

[0080] 其中,  $D$ 表示所述AGV本循环从起点到终点在二维栅格地图中行驶的总欧氏距离;  $u$ 表示AGV行驶的总栅格数, 包含起点和终点一共经过 $u+1$ 个栅格;  $(x_j, y_j)$ 表示所述AGV经过的第 $j$ 个栅格坐标;

[0081] AGV最优路径保证在路径长度最短的前提下, 转弯数最少:

$$[0082] \quad d = \begin{cases} D, & D < d \\ d, & D > d \end{cases}$$

$$[0083] \quad c = \begin{cases} \mathit{corner}_{\text{总}}, & \mathit{corner}_{\text{总}} < c \\ c, & \mathit{corner}_{\text{总}} > c \end{cases}$$

[0084] 其中,  $D$ 表示本循环AGV从起点到终点的总欧氏距离;  $d$ 表示已知的AGV从起点到终点的最短欧氏距离;  $\mathit{corner}_{\text{总}}$ 表示本循环AGV从起点到终点的转弯数;  $c$ 表示已知的AGV从起点到终点的最少转弯数; 最短欧氏距离 $d$ 和最少转弯数 $c$ 的初始设置值均为 $+\infty$ 。

[0085] 本发明的另一个具体实施例, 公开了一种基于DQN的AGV全局路径规划系统。参考图10, 基于DQN的AGV全局路径规划系统包括: 栅格地图构建模块1002, 用于基于感知到的仓储环境信息构建栅格边长为 $p$ 的二维栅格地图的模拟环境, 其中, 仓储环境信息包括障碍物、AGV的起点、AGV的当前位置及AGV的终点位置; 动作集扩充模块1004, 用于扩充AGV在二维栅格地图中下一步采取的动作集; 奖励函数设置模块1006, 用于设置连续合理的奖励函数以区别AGV在二维栅格地图中每个栅格处获取的奖励值; 以及最优路径输出模块1008, 用于设计最优路径判别条件并根据最优路径判别条件在AGV运行完指定循环后, 输出最优路径。

[0086] 下文中, 参考图2至图9以具体实例的方式, 对基于DQN的AGV全局路径规划方法进行详细描述。

[0087] 参考图2, 对基于DQN的AGV全局路径规划方法可以包括:

[0088] S1、对环境信息进行感知, 获取障碍物位置、AGV的起点、AGV的当前位置及AGV的终点位置, 构建栅格边长为 $p$ 的二维栅格地图模拟环境, 具体地,  $p$ 为二维栅格地图中的单位栅

格的边长。

[0089] S2、为了增加路径规划的准确性,扩充AGV动作集,将传统DQN在栅格图中下一步能采取的动作由4个扩充为8个。

[0090] S3、为了缩短基于DQN的AGV全局路径规划时间,提升路径规划的效率,设置连续且合理的奖励函数reward,保证AGV在每一个栅格获取奖励值的区别,减少AGV无效探索的时间。

[0091] S4、为了配合AGV动作集的扩充,设计最优路径判别条件,根据此判别条件,在AGV运行完指定的回合(即,指定次数的循环)后,输出一条最优路径。

[0092] 参考图9,具体地,S1:构建栅格边长为 $p$ 的二维栅格地图模拟环境。

[0093] 仿真环境采用pycharm编写,大小为 $17 \times 25$ 的栅格地图。图6中左上角黑色圆圈AGV等待区,下方浅灰色圆圈代表卸货点,右上方深灰色圆圈代表上包处,其余黑色方块代表障碍物;图7a、图8a中黑色圆圈代表AGV起点,浅灰色圆圈代表AGV终点,白色小圆圈代表AGV路径,黑色方块代表障碍物;图7b、图7c、图8b中浅灰色圆圈代表AGV起点,深灰色圆圈代表AGV终点,白色小圆圈代表AGV路径,黑色方块代表障碍物。

[0094] S2:扩充AGV动作集;

[0095] 将传统DQN在栅格图中下一步能采取的动作由4个扩充为8个,参考图4a,现有的AGV动作集只包括上、下、左、右共4个方向,参考图4b,扩充AGV动作集为上、下、左、右、左上、右上、左下、右下共8个方向。

[0096] AGV能通过执行动作集中的动作,到达8个临近栅格中的一个。

[0097] AGV在当前位置坐标 $(x_i, y_i)$ 通过执行动作 $a$ ,就可到达栅格图中的临近位置坐标 $(x_{i+1}, y_{i+1})$ :

$$[0098] \quad (x_{i+1}, y_{i+1}) = \begin{cases} (x_i, y_i + p), & a = up \\ (x_i, y_i - p), & a = down \\ (x_i - p, y_i), & a = left \\ (x_i + p, y_i), & a = right \\ (x_i - p, y_i + p), & a = upleft \\ (x_i + p, y_i + p), & a = upright \\ (x_i - p, y_i - p), & a = downleft \\ (x_i + p, y_i - p), & a = downright \end{cases}$$

[0099] 其中, $(x_i, y_i)$ 表示所述AGV当前位置坐标, $(x_{i+1}, y_{i+1})$ 表示所述AGV执行完动作集中动作 $a$ 后所处的邻近位置坐标; $(x_i, y_i + p)$ 和 $(x_i, y_i - p)$ 分别表示在所述AGV当前所在位置坐标基础上, $y$ 轴正和负方向添加一个栅格边长; $(x_i + p, y_i)$ 和 $(x_i - p, y_i)$ 分别表示在所述AGV当前所在位置坐标基础上, $x$ 轴正和负方向添加一个栅格边长。

[0100] S3:设置连续且合理的奖励函数reward。

[0101] 根据Q-learning算法,建立Q值表,记录AGV与环境不断交互得到的奖励值,不断迭代修改Q表,增加选择正奖励的概率,进而不断更新动作策略集,使AGV每一步动作都趋向于

最优动作。Q-learning算法的基本形式为：

$$[0102] \quad Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t) \right]$$

[0103] 其中,  $r_{t+1}$  代表状态  $s_t$  情况下执行完动作  $a_t$  后立刻得到的奖励,  $s_{t+1}$  表示  $s_t$  执行完动作  $a_t$  后的下一状态,  $a$  则代表状态  $s_{t+1}$  情况下采取的所有动作,  $\gamma$  代表对未来奖励的衰减因子; 左侧的  $Q(s_t, a_t)$  代表修改后的Q值表, 右侧的  $Q(s_t, a_t)$  代表Q值表中已有的Q估计值,  $r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a)$  是Q现实值,  $\alpha$  代表学习率, 对所述Q估计值和所述Q现实值之间有多少误差进行学习。

[0104] DQN中则构建两个神经网络, 通过神经网络来代替Q表, 一个用于计算Q估计, 另一个用于计算Q现实, 采用二者的均方差来训练网络, 更新网络的损失函数。DQN自身包括两个神经网络(两个Q网络), 都能输出Q值, 但有一个网络可以被称为target Q网络, 其与另一个Q网络结构一样, 初始的权重也一样, 只是Q网络每次迭代都会更新, 而target Q网络是每隔一段时间才会更新, 权重更新较慢, 因此权重记为  $\omega^-$ 。target Q网络计算Q值用于产生误差用于学习、辅助网络权重的更新, Q网络输出的值决定Q值, 以及下一步AGV坐标。

$$[0105] \quad L_t = \left[ r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a; \omega^-) - Q(s_{t+1}, a; \omega) \right]^2$$

$$[0106] \quad \omega_{t+1} = \omega_t + L_t * \nabla Q(s_{t+1}, a; \omega)$$

[0107] 其中,  $s_t$  为  $t$  时刻AGV的状态,  $Q(s_t, a_t)$  为在  $s_t$  状态下采取动作  $a_t$  所取得的值,  $\max_{a \in A} Q(s_{t+1}, a)$  为  $s_{t+1}$  状态下采取所有动作  $a$  的Q值最大者,  $r_{t+1}$  是对状态  $s_{t+1}$  的评估, 表示AGV由  $s_t$  状态执行动作到  $s_{t+1}$  状态所能取得的奖励值;  $\gamma$  为折扣因子, 表示未来奖励对当前动作的影响程度;  $Q(s_{t+1}, a; \omega^-)$  为预测网络Q值,  $Q(s_{t+1}, a; \omega)$  为目标网络Q值;  $\omega^-$ 、 $\omega$  为神经网络相关参数;  $\nabla$  为梯度符号。DQN利用神经网络逼近  $Q(s_t, a_t)$ , 并通过梯度下降最小化误差。

[0108] 参考图3, DQN中两个神经网络具有不同的网络参数, 但是具有完全相同的网络结构, 每个网络由输入层、隐藏层、输出层构成, 输入层输入AGV当前状态  $S_i(x, y)$ , 隐藏层含有  $n$  个神经元, 激励函数选择RELU函数, 输出动作集中动作的Q值, 根据  $\epsilon$ -greedy 原则做出决策, 选择下一步AGV所要采取的动作。

[0109] AGV每做出一个动作会获得一个奖励值, 为了获得更高的奖励, AGV会在动作集中选取合理的动作。考虑到在后续局部路径规划中因转弯角度而带来的兜圈绕路问题, 为了节约AGV能耗, 降低AGV故障及零件磨损概率, 在终点前的奖励函数中加入转弯数  $corner$ , 减少AGV的转弯数, 实现最优路径规划。

[0110] 设置AGV在栅格图中下一步的奖励函数, 如下所示:

$$[0111] \quad reward = \begin{cases} -1, & \text{预计下一步遇到障碍物} \\ 1 - m * \left( \frac{step\ distance}{long\ distance} \right) + 1 - n * corner, & \text{预计下一步遇到目标点} \\ 1 - \frac{next\ distance}{cur\ distance}, & \text{其他} \end{cases}$$

[0112] 其中,  $step\ distance$  表示预计本回合到达终点走过的总欧氏距离,  $long$

$distance$ 表示之前所有回合走到终点所走过的最长欧氏距离, $corner$ 表示本回合到达终点所进行的转弯数, $cur\ distance$ 表示这一步走完到达终点的欧氏距离, $next\ distance$ 表示下一步走完到达终点的欧氏距离。 $m$ 、 $n$ 表示常数系数,在实例中取 $m=0.9$ , $n=0.01$ 。

[0113] 对AGV上一步、当前、下一步共三个栅格的坐标进行计算,判断AGV是否出现转弯情况,如果不满足走直线的要求,则说明AGV出现了转弯。

[0114] 符合AGV直线运动的坐标表达式如下所示:

[0115] 沿水平直线方向运动:

$$[0116] \quad (x_{i-1} - p, y_{i-1}) = (x_i, y_i) = (x_{i+1} + p, y_{i+1})$$

$$(x_{i-1} + p, y_{i-1}) = (x_i, y_i) = (x_{i+1} - p, y_{i+1})$$

[0117] 沿竖直直线方向运动(参考图5a):

$$[0118] \quad (x_{i-1}, y_{i-1} - p) = (x_i, y_i) = (x_{i+1}, y_{i+1} + p)$$

$$(x_{i-1}, y_{i-1} + p) = (x_i, y_i) = (x_{i+1}, y_{i+1} - p)$$

[0119] 沿对角线方向运动(参见图5b):

$$[0120] \quad (x_{i-1} - p, y_{i-1} - p) = (x_i, y_i) = (x_{i+1} + p, y_{i+1} + p)$$

$$(x_{i-1} - p, y_{i-1} + p) = (x_i, y_i) = (x_{i+1} + p, y_{i+1} - p)$$

$$(x_{i-1} + p, y_{i-1} + p) = (x_i, y_i) = (x_{i+1} - p, y_{i+1} - p)$$

$$(x_{i-1} + p, y_{i-1} - p) = (x_i, y_i) = (x_{i+1} - p, y_{i+1} + p)$$

[0121] 其中, $(x_i, y_i)$ 表示AGV当前所在位置坐标, $(x_{i+1}, y_{i+1})$ 表示AGV执行完动作后所处的位置坐标; $(x_{i-1}, y_{i-1})$ 表示AGV上一步所在位置坐标; $(x_i, y_i+p)$ 表示在AGV当前所在位置坐标基础上, $y$ 轴方向添加一个栅格边长; $(x_i+p, y_i)$ 表示在AGV当前所在位置坐标基础上, $x$ 轴方向添加一个栅格边长。

[0122] S4:设计最优路径判别条件

[0123] 统计成功到达终点的AGV各回合总路径,统计它们从第一步到倒数第二步通过的坐标;

[0124] 对AGV上一步、当前、下一步共三个栅格的坐标进行计算,判断AGV是否出现转弯情况;

[0125] 如果不满足走直线的要求,则说明AGV出现了转弯,则从起点开始的整个路径转弯数加一: $corner = corner + 1$

[0126] 其中, $corner$ 代表本回合路径中已有的转弯数。

[0127] 通过判断AGV在栅格图中实际行驶的欧氏距离及最终路径转弯数,确定本回合内所能得到的最优路径。



[0128] AGV从起点到终点在栅格图中的实际欧氏距离：

$$[0129] \quad D = \sum_{j=1}^u \sqrt{(y_j - y_{j-1})^2 + (x_j - x_{j-1})^2}$$

[0130] 其中, $D$ 表示AGV本循环从起点到终点在二维栅格地图中行驶的总欧氏距离; $u$ 表示AGV行驶的总栅格数,包含起点和终点一共经过 $u+1$ 个栅格; $(x_j, y_j)$ 表示AGV经过的第 $j$ 个栅格坐标;

[0131] AGV最优路径需要保证在路径长度最短的前提下,转弯数最少:

$$[0132] \quad d = \begin{cases} D, & D < d \\ d, & D > d \end{cases}$$

$$[0133] \quad c = \begin{cases} \text{corner}_{\text{总}}, & \text{corner}_{\text{总}} < c \\ c, & \text{corner}_{\text{总}} > c \end{cases}$$

[0134] 其中, $D$ 表示本循环AGV从起点到终点的总欧氏距离; $d$ 表示已知的AGV从起点到终点的最短欧氏距离; $\text{corner}_{\text{总}}$ 表示本循环AGV从起点到终点的转弯数; $c$ 表示已知的AGV从起点到终点的最少转弯数;最短欧氏距离 $d$ 和最少转弯数 $c$ 的初始设置值均为 $+\infty$ 。

[0135] 参考图9,AGV到达下一步坐标后,判断AGV是否到达终点,当没有到达终点时,进一步判断是否碰到障碍物。如果AGV没有碰到障碍物则更新地图并进入下一次循环;否则当AGV碰到障碍物时则结束。当AGV到达终点时,判断当前路径是否优于之前的最优路径。如果当前路径优于之前的最优路径,则当前路径为最优路径,否则保留之前的最优路径。

[0136] 当运行完设置的5000回合后,选择AGV路径欧氏长度最短或者转弯数最少的路径作为最优路径。

[0137] 本申请的核心是提供一种全局路径规划方法,通过合理的奖励函数有效提高路径规划的效率,保证仓储中心的高效工作;本申请的另一核心是提供一种高效的最优路径判别条件,配合增加的AGV运动方向,有效提升路径规划的准确率。

[0138] 本领域技术人员可以理解,实现上述实施例方法的全部或部分流程,可以通过计算机程序来指令相关的硬件来完成,所述的程序可存储于计算机可读存储介质中。其中,所述计算机可读存储介质为磁盘、光盘、只读存储记忆体或随机存储记忆体等。

[0139] 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揭露的技术范围内,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。

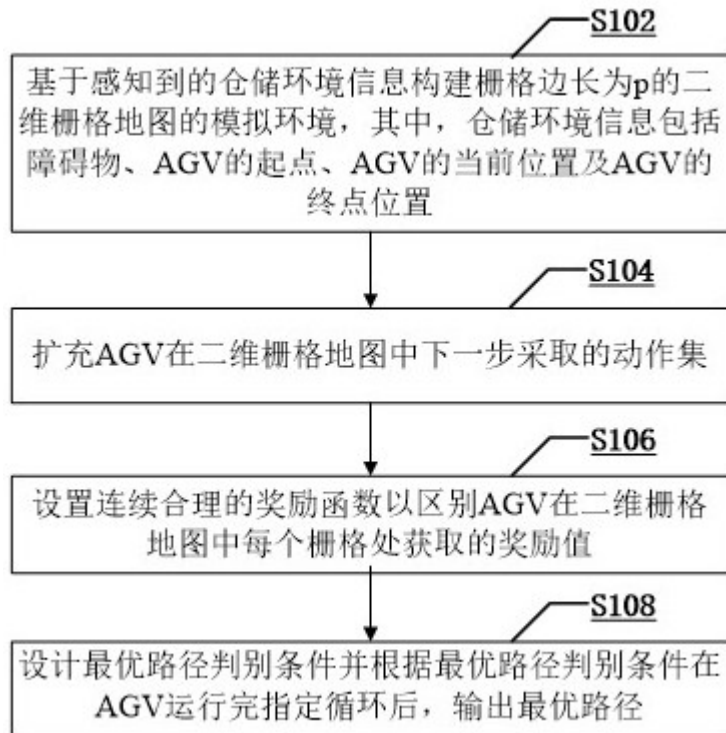


图1

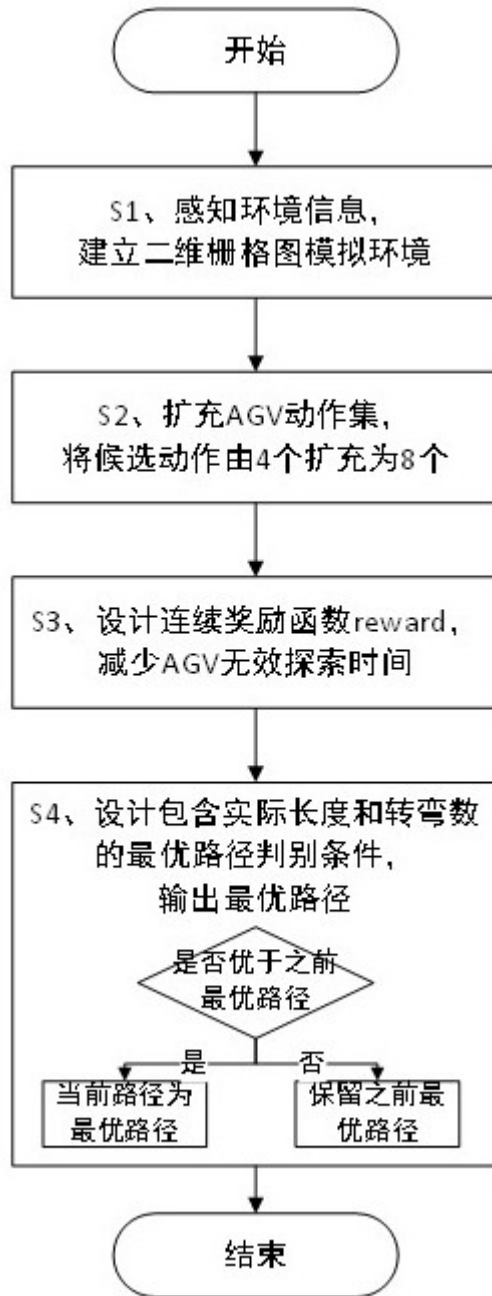


图2



图3

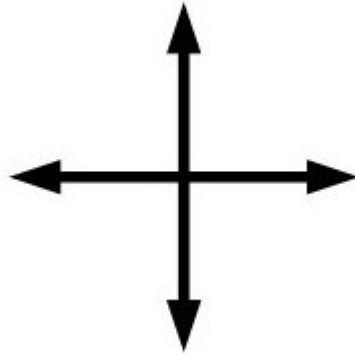


图4a

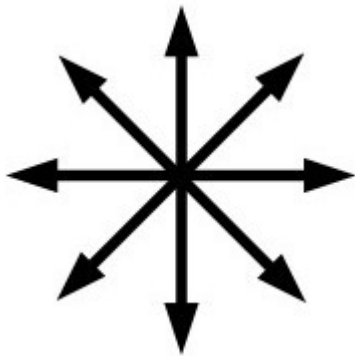


图4b

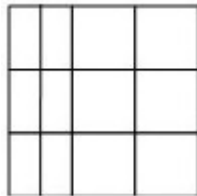


图5a

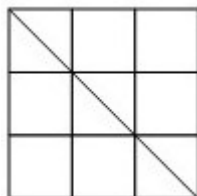


图5b

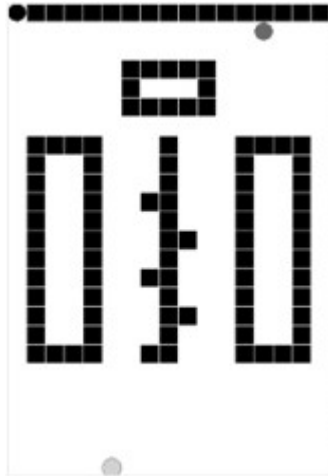


图6

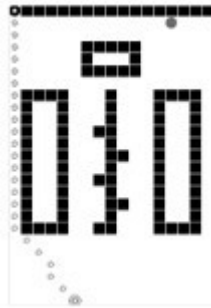


图7a

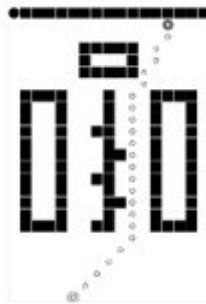


图7b

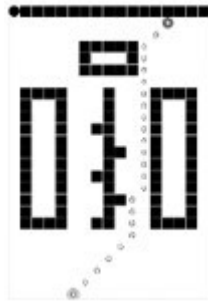


图7c

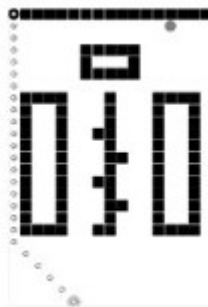


图8a

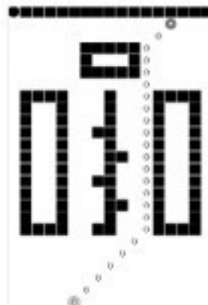


图8b

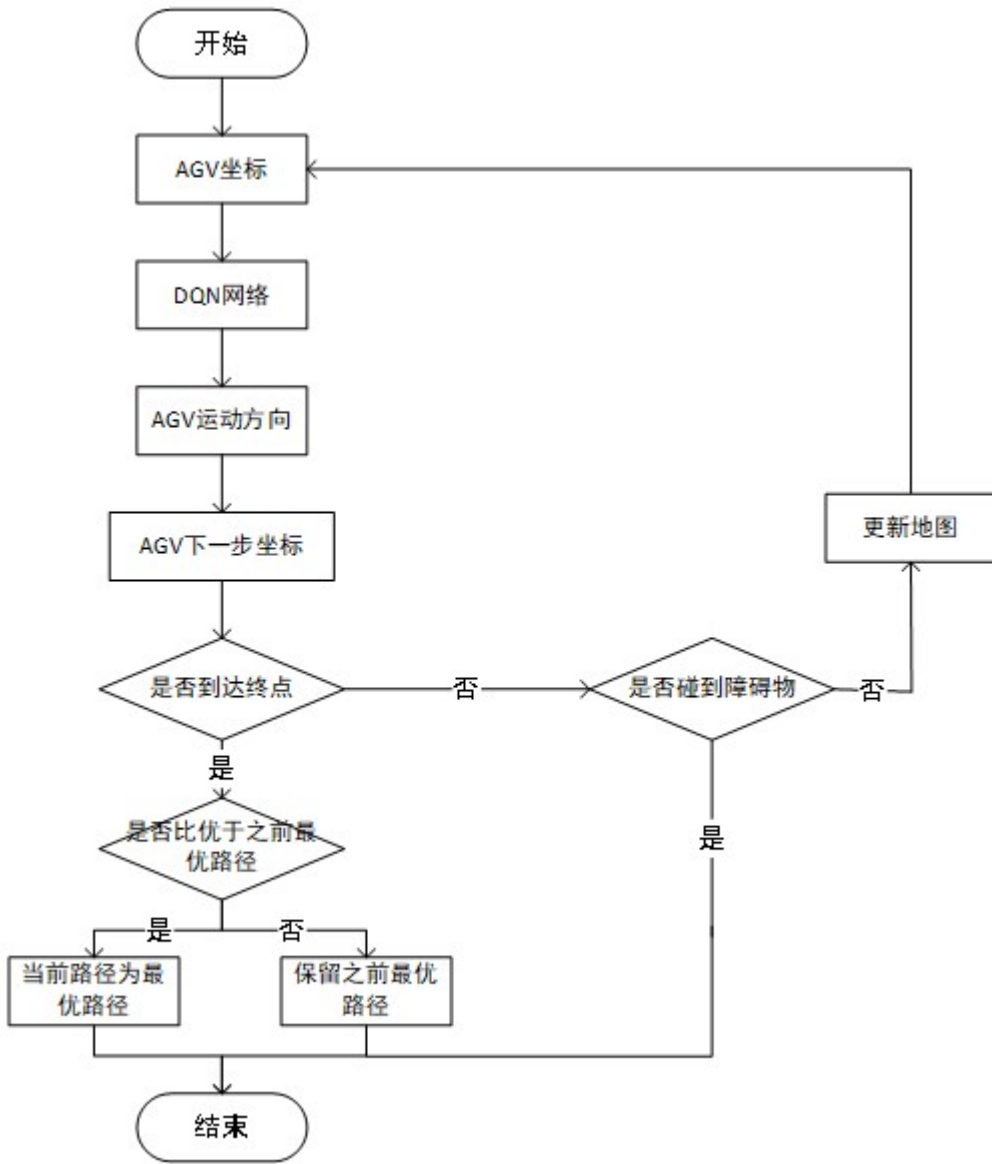


图9

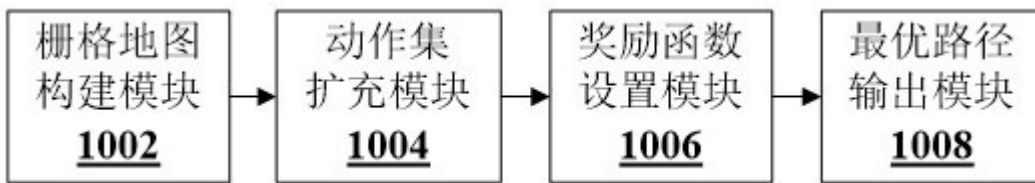


图10