**建筑工地环境下机器人室内路径规划算法专利交底书**

**一、背景技术（最接近的现有技术）**

|  |
| --- |
| 发明的技术背景，或者别人是怎么做的，存在什么缺点？  建筑工艺种类繁多并且各种工艺覆盖范围差异很大，由于施工环境复杂多变，并不能完全保证每一个施工任务刚好位于特定的位置，有可能在地图上的任意位置，因此对机器人实现可以任意点路径规划功能就显得格外重要。  现有的路径规划算法，例如：Dijkstra算法、A\*算法，都是基于固定路径下并且可规划的点只能是在路网上的点，基于该点寻找到目标点最短路径，偏于理想情况，在实际情况下机器人可能起始位置偏离路网上的点，这时机器人需定位回到起始点花费很长时间，甚至是从该点出发会碰到墙体，使得机器人运输效率降低并且安全性大大降低，并且Dijkstra路径规划算法搜寻最短路径所花费时间较长，并不适用于实时路径规划。  显然目前算法并不适用于任意点路径规划，我们提出的搜寻当前最近路网点并基于该点再使用A\*算法进行最优路径规划，从而得到任意点最优路径规划，不仅实现路径最优，并且可以实现任意点进行施工任务接取以及进行施工作业。 |

**二、主要改进点及解决的技术问题**

|  |
| --- |
| 1. 最主要/关键的技术问题   任意点进行路径规划不碰到周围墙体   1. 次要问题（如有多个，依次列出）   最短路径搜寻  判断两点之间是否有墙体 |

**三、技术效果（详细阐述）**

|  |
| --- |
| 1.机器人从任意位置可以进行任务接取  2.机器人从起点位置经由最短路径并且无碰撞到达路网上的点  3.机器人通过路网进行最优路径规划到达任意任务地点 |

**四、具体实施方案以及附图（重点部分，详细阐述）**

|  |
| --- |
| 需要注意：   1. 不能只有原理描述或仅作功能介绍，每一功能都应有相应的实现方案。 2. 图文结合详细阐述具体的实施方式（组成部件+部件连接关系+工作过程/原理+对应的技术效果），力求使阅读者快速明白技术方案内容，可配合结构图、流程图、原理框图、电路图、时序图等进行说明。 3. 英文缩写应有中文注释。 4. 产品专利需提供CAD结构图，并标记各部件名称；控制、软件、业务方法，需提供流程图。   **说明：路网是基于施工场地构建的点到点的路径地图，其中点的选取是为了更好实现多机资源管控防止冲突以及死锁问题，也是为了防止机器人行进过程中碰撞墙体。**  1.机器人每次收到任务指令，从起点移动到终点，都会调用一次任意点路径规划算法，路径规划方案总体逻辑如下：  **将起点到终点的总路径长度total分为三段**  **1.起点到路网起点startpoint的距离distance1**  **2.路网起点startpoint到路网终点endpoint的距离length**  **3.路网终点endpoint到终点的距离distance2**  **步骤如下：**  S1：路网外点（X1，Y1）到路网上某点（X2，Y2）的距离公式：S = |(x2-x1)|+|(y2-y1)|  S2：首先判断起点和终点是否在墙体内部，若在墙体内部，则提示路径错误，若都不在墙体内部，则继续S3。  S3：通过循环遍历判断起点是否是路网上的某一个点，即起点坐标是否与路网某个点相同，若相同，则将该路网点设为startpoint，distance1 = 0,startlist = [startpoint],跳转至S7，若遍历完所有路网点没有相同点，则继续S4。其中，startlist代表可选择的路网起点的列表。  S4: 通过距离公式计算起点到同层楼范围内各点的距离S，并添加到distancelist1中，最后对distancelist1中到各点距离进行从小到大排序得到sortlist1。  S5：从排完序的列表sortlist1中从前往后每次取出一个点，也就是根据到起点距离从小到大的原则每次取出一个点。  S6：判断起点到该点路径上是否有障碍物，若有障碍物，跳转至S5，若无障碍物，把该点存至startlist中，继续S5直至遍历完所有路网点。  S7：同理循环遍历判断终点是否为路网上的某一个点，若是，则将该路网点做endpoint，distance2=0，endlist=[endpoint],跳转至S11，若遍历完路网点没有相同点，则继续S8。  其中，endlist代表可选择的路网终点的列表。  S8：同理通过距离公式计算终点到同层楼范围内各点的距离S，并添加到distancelist2中，最后对distancelist2中到各点距离进行从小到大排序得到sortlist2。  S9：从排完序的列表sortlist2中从前往后每次取出一个点，也就是根据到起点距离从小到大的原则每次取出一个点  S10：判断起点到该点路径上是否有障碍物，若有障碍物，跳转至S9，若无障碍物，保存该点至endlist中，继续S9至遍历完所有路网点。  S11：双层循环分别取出startlist和endlist中各一个点作为startpoint和endpoint。  S12：根据startpoint和endpoint用传统A\*算法进行路网最短路径规划，将路网最短路径规划的距离设为length。  其中：总距离=length + startpoint与任起点的距离distance1 + endpoint与终点的距离distance2  将总距离保存至total列表中，继续S11直至循环完成  S13：从total中取出总距离最小的距离即为最短路径，当total最小距离有多条路径时，会选择distance1和distance2值之和最小的路径作为最优路径，从而实现最短路径规划。  **通过使用本发明，可以实现以下效果：**  在建筑工地上，需要机器人执行室内移动任务，到达地图上任一点时，机器人可以使用本发明寻找可行的最短路径到达，实现建筑工地环境下机器人室内任意点移动的功能。  **具体算法实施方案：**  **S1: 传统A\*算法的公式**：f=g+h，构造启发式函数：g = G + W。其中G为从起点到当前节点的总时间。w为从起点到当前节点的等待时间之和。构造启发式函数：h = H 。其中H为当前节点到目标节点通过Dijkstra算法得出的最短距离所对应的时间，通过该算法来计算规划路网上startpoint到endpoint的路径  **S2：判断起点终点是否在墙体内部算法原理：**  S201：将墙体分解成多个矩形范围，通过分别判断起点和终点是否在矩形内或者矩形边界上，从而来判断起点和终点是否在墙体内部  S202：连接点与矩形的四个顶点，形成四个三角形，若点在矩形内部或边界上，那么必定有四个三角形的面积之和等于矩形面积，反之，则四个三角形的面积之和不等于矩形面积。  **具体情景如下：**  (x1, y2)  (x3, y3)  (x2, y2)  (x1, y1)  (x2, y1)  如上图所示：若某点的坐标为（x3，y3），当前进行判断的墙体矩形4个点坐标分别为  (x1，y1)、(x1, y2)、(x2, y1)、(x2, y2),这时候要判断该点是否在矩形内或边界上。则可以根据上述算法进行判断。  若三角形三个顶点分别为(x1，y1)、(x2，y2)、(x3，y3)：  则该三角形面积：S1 = |(x2-x1)(y3-y1)-(x3-x1)(y2-y1)| / 2  矩形面积：S2 = （x2-x1）(y2-y1)  四个三角形面积之和 S’若与矩形面积相等，则点在墙体矩形内部或边界，程序提示路径信息错误，反之，则点不在墙体矩形，则可以进行路径规划。  **S3：判断两点之间是否有墙体的算法原理**：  S301：将墙体分解成多个矩形后，将矩形外轮廓表示为4条线段，根据实际情况任意点到路网点需做平行于x轴与平行于y轴的两条线段，则两点的路径可分为两种情况：  （x2,y2）  （x1,y1）  如上图所示：从左下角的点(x1,y1)到右上角的点(x2,y2)，则有两条路径到达  1.先沿着水平方向移动，到达x = x2时，再沿着竖直方向移动到达目标点（如绿色箭头所示）  2.先沿着竖直方向移动，到达y = y2时，再沿着水平方向移动到目标点（如红色箭头所示）  **S302：判断路径是否穿过墙体**  （x2,y2）  （x1,y1）  如上图所示：  将一块矩形墙体轮廓表示为四条线段，两条竖直线段，两条水平线段，并且竖直线段和水平线段分别构造成一个字典  1.竖直墙体线段字典以X坐标值为键，该键对应的值为一个列表,{x1:[y1,y2,y3,y4]},代表在X坐标值为x1时，y1至y2的线段以及y3至y4的线段都是墙边界。  例：{40:[25,55,90,180]}表示在横坐标x=40时，有两条线段为墙边界，分别为纵坐标y=25至y=55以及y=90至y=180.  2.水平墙体线段字典以Y坐标值为键，该键对应的值为一个列表,{y1:[x1,x2,x3,x4]},代表在Y坐标值为y1时，x1至x2的线段以及x3至x4的线段都是墙边界。  **S303：需判断的路径有两条，每条路径都需判断水平路径和竖直路径是否穿过墙体**  1.水平路径判断：只需要筛选出竖直线段字典中键的大小在（x1,x2）之间的键值对，并判断水平路径的y值是否在筛选出墙的边界线段范围内，若在，则表示穿过了墙体，路径不可行。反之，可行  2.竖直路径判断(原理一样)：只需要筛选出水平线段字典中键的大小在（y1,y2）之间的键值对，并判断竖直路径的x值是否在筛选出墙的边界线段范围内，若在，则表示穿过了墙体，路径不可行。反之，可行  **S304：实例说明**  例：如下图，左下起点坐标为(40, 40)，右上路网点坐标为(100, 80)，将墙体边界坐标转换为上述格式  竖直墙体边界字典为: {x=20:[60,120], x=30:[60,120], x=70:[60,70]}}  水平墙体边界字典为：{y=60:[20,70], y=70:[30,70], y=120:[20,30]}  **1.判断实线路径：**  **实线路径的水平路径1：**  已知两点生成的水平路径1为x=40至x=100，其y值为40，此时从竖直墙体边界字典中筛选出x处于（40，100）区间内的键值对为x=70:[60,70],该水平路径1的y=40，不在键值对所对应的[60,70]区间内，所以该水平路径1不会碰撞墙体。  **实线路径的竖直路径2：**  已知两点生成的竖直路径2为y=40至y=80，其x值为100，此时从水平墙体边界字典中筛选出y处于（40，80）区间内的键值对为y=60:[20,70]以及y=70:[30,70],该竖直路径2的x=100，不在[20,70]和[30,70]任意一个区间内，所以该竖直路径2不会碰撞墙体。  因此实线整体路径不会碰撞墙体  **2.判断虚线路径：**  判断原理跟实线路径一样，通过判断发现虚线路径的竖直路径3会碰撞墙体。  4  3  2    1    通过上述可知图中实线路径可行，虚线路径不可行。 |

**五、其他拓展方案**

|  |
| --- |
| 1、说明哪些技术手段是必不可少的？可以省略的？其他替代方案？  2、是否还有其他应用场景/场合/领域？  **将墙体的外轮廓表示为四条边线，可以大大简便运算以及减少路径规划时间。**  **计算起点和终点是否在墙体内部必不可少的原因：若起点或终点在墙体内，将始终找不到合适的路径进行任务运输。**  **计算两点之间是否有墙体必不可少：若不进行计算，机器人将有可能直接撞向墙体，造成安全事故以及损失。**  **此算法还能应用于任意点运输作业体系，例如：货物运输系统** |

**六、重要技术创新主体提醒**

|  |
| --- |
| 借鉴的技术文献，主要同行竞争对手、科研机构信息、专家名称等。 |