

## 基于A\*算法的通用仓储物流 三维路径规划研究

Research on 3D path planning of universal  
warehousing logistics based on A \* algorithm

方 华, 周奇才, 熊肖磊, 赵 炯

FANG Hua, ZHOU Qi-cai, XIONG Xiao-lei, ZHAO Jiong

(同济大学 机械与能源工程学院, 上海 201800)

**摘 要:** 仓库作为物流转运的中心, 追求低错单率和低人工成本, 提高仓储系统的整体运转效率, 成为其发展的主要目标。拣选机构跨层作业作为一种新的解决方案, 提高了仓储系统的效率也提高了系统灵活性, 使得仓储物流路径规划的约束变少。传统的仓储物流路径规划局限于某一特定仓库, 设计一种相对适用的路径规划模型, 且多为二维路径规划。但不同的仓库结构要重新设计算法, 模型利用率不高。跨层作业此类约束较少甚至无约束的三维仓储路径规划研究不多。文章基于栅格化的仓储环境, 提出一种通用的仓库拣选三维路径规划模型, 从设备能耗、拣选时间两方面考虑设计目标函数, 拣选设备水平、竖直方向运动特性不同, 设计分段启发函数的A\*算法进行求解, 并用matlab进行仿真, 验证了模型的可行性。

**关键词:** 仓储物流; 三维路径规划; A\*算法; 通用模型

**中图分类号:** TP29

**文献标识码:** A

**文章编号:** 1009-0134(2018)07-0026-06

### 0 引言

跨层作业逐渐成为仓储物流提高效率的趋势之一。ADAPTO是范德兰德公司为了减轻企业仓储配送压力而专门设计的一款创新的3D穿梭小车。ADAPTO逐渐弱化了仓储系统层与层、巷道与巷道之间的隔离, 同一个3D穿梭车可以自由的穿梭于货架任意位置, 不局限于巷道与层之间。同样, 瑞士的建筑五金供应商SFS采用的跨层穿梭车系统, 立库有17层, 这17层仅配备了3台穿梭车, 共5个巷道, 整个立库仅用15台穿梭车, 系统设计不仅降低成本, 作业效率也非常高<sup>[1]</sup>。跨层作业极大地增加了拣选作业的灵活性。

具有3维空间运动能力的典型移动机器人, 现在研究较多的是各类空间和水下机器人。其中, 智能程度较高的包括MAV、UAV、UUV、AUV等<sup>[2]</sup>。智能仓储物流领域使用拣选机器人十分广泛, 但大部分路径规划研究局限于二维路径规划, 类似跨层作业此类约束较少甚至无约束的通用仓储模型三维路径规划少有研究涉及, 本文提出一种通用的仓库拣选三维路径规划模型, 并设计了与之匹配的寻优算法。

### 1 问题分析

#### 1.1 仓储环境描述

通用立体仓库模型中的拣选小车路径规划问题是一

个三维路径规划, 目前研究较多的是无人机(UAV)的三维路径规划, 仓储环境下的全地图三维路径规划研究并不多见。无人机三维路径规划的环境威胁包括地图环境威胁、雷达侦查威胁、敌军导弹辐射范围威胁等。仓储拣选设备的路径规划与之对比, 显然仓库里的拣选车辆路径比较规则, 通常只有水平、竖直直线运动, 相比于无人机的飞行轨迹较为简单, 一个无人机的路径选择在栅格空间中可以有26个方向<sup>[3]</sup>(如图1所示)。一个仓储机器人的路径选择在空间中通常只有6个方向, 如图2所示。

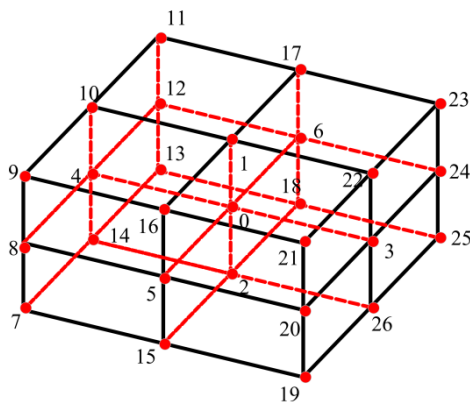


图1 无人机(UAV)运动方向

收稿日期: 2018-02-02

作者简介: 方华(1994-), 男, 江苏人, 硕士研究生, 研究方向为现代物流技术装备控制及自动化。

【26】 第40卷 第7期 2018-07

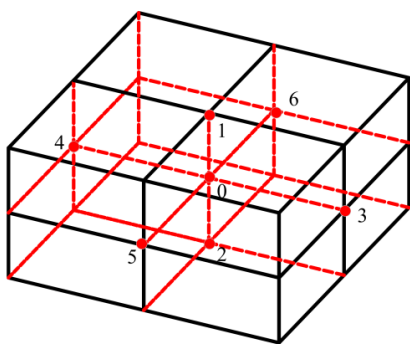


图2 仓储机器人运动方向

仓库拣选设备（提升机、堆垛机、穿梭车等）的运动仅为前、后、左、右、上、下6个方向。分别对应矩阵：

$$M = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} \quad (1)$$

故将连续空间变换为栅格空间系统模型，更符合立体仓库拣选设备运动的客观规律，同时便于算法搜索。且实际求解过程中仓库尺寸、规模已知，因此可以视为全局路径规划问题。将仓库根据每个货位的尺寸按小方块进行划分，每个小立方体代表一个货位。

## 1.2 A\*算法距离选择

A\*算法是一种常见的路径搜索算法，A\*算法的估价函数可以表示为：

$$f(n) = g(n) + h(n), h(n) < h^*(n) \quad (2)$$

其中 $g(n)$ 表示从起始节点Start到节点 $n$ 的真实耗费值， $h(n)$ 表示从节点 $n$ 到终止点Goal的启发估计耗费值。 $f(n)$ 表示从起始点开始，经过节点 $n$ 到达目标的启发估计耗费值。 $h^*(n)$ 是指经过节点 $n$ ，到达终止目标点的实际最优消耗值。其基本思路与经典的路径搜索算法Dijkstra相似，但在Dijkstra算法的基础上加上启发代价，路径代价通常由距离决定，距离通常有三种方式计算：

### 1) 曼哈顿距离

曼哈顿距离即坐标系中两点的绝对轴距之和。其表达式如式(3)所示，效果如图3所示。

$$h_M(n) = (|x_n - x_{goal}| + |y_n - y_{goal}|) \quad (3)$$

### 2) 切比雪夫距离

切比雪夫距离被称为棋盘距离。其表达式如式(4)所示，效果如图4所示。

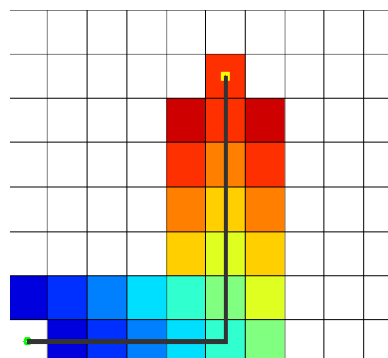


图3 曼哈顿距离的A\*搜索过程

$$h_C(n) = \max(|x_n - x_{goal}|, |y_n - y_{goal}|) \quad (4)$$

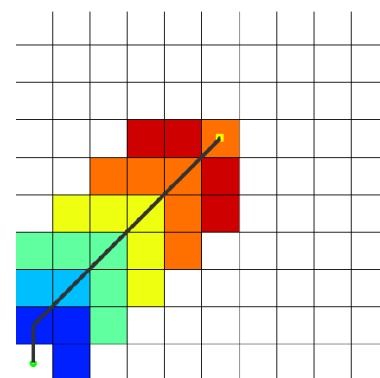


图4 切比雪夫距离的A\*搜索过程

### 3) 欧几里得距离

欧几里得距离是衡量两点之间距离远近的最常用的方法之一，其值直接可以看作两个位置点在欧式空间中的两点的直线距离，其表达式如式(5)所示，效果如图5所示。

$$h_E(n) = \sqrt{(x_n - x_{goal})^2 + (y_n - y_{goal})^2} \quad (5)$$

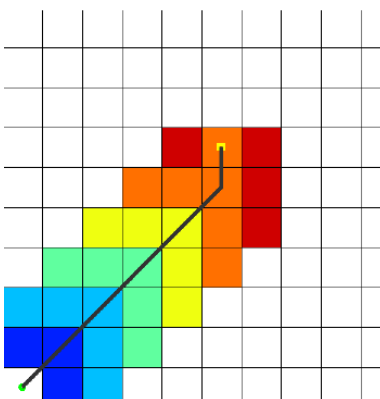


图5 欧几里得距离的A\*搜索过程

由于仓储环境中，只存在直线运动且为前、后、左、右、上、下6个方向，因此算法的距离选择应以曼哈顿距离为基础，在此距离公式基础上增加竖直方向坐标差的绝对值，距离公式为：

$$h_{3D}(n) = |x_n - x_{goal}| + |y_n - y_{goal}| + |z_n - z_{goal}| \quad (6)$$

## 2 模型建立

### 2.1 参数归一化处理

线性函数归一化（Min-Max Scaling），也称为离差标准化，是对原始数据的线性变换，使结果值映射到[0-1]之间。转换函数如下：

$$x^* = \frac{x - \min}{\max - \min} \quad (7)$$

其中max为样本数据的最大值，min为样本数据的最小值。本文A\*算法的历史路径函数里还包括地图环境代价，环境代价的取值为0或1，在计算代价函数时，为避免“大数吃小数”的现象，需要对数据进行归一化处理，本文涉及的标准化参数有单位距离设备的能耗和单位距离设备的拣选时间两个指标，其中单位距离设备的拣选时间与设备运行速度有关。

统计出仓储系统内不同执行设备的单位距离能耗和速度，找出能耗最大值和速度最大值，根据以上离散标准化公式将单位能耗和速度归一化处理，使之映射到[0-1]之间，由于系统考量的两个指标是单位能耗与拣选时间，时间与速度成反比，因此在速度值的归一化处理之后可作加一处理，避免拣选时间数值无穷大，使单位距离下的设备运行时间也在[0-1]之间。

假设纵向移动由提升机执行，其速度为 $v_x$ 、单位能耗（功率）为 $E_x$ ，水平移动由穿梭车执行，其速度为 $v_y$ 、单位能耗（功率）为 $E_y$ 。则有：

$$\{O_x, P_x\} = \{\min, \max\{E_{x1}, E_{x2}, E_{x3} \cdots E_{xn}\}\} \quad (8)$$

$$\{O_y, P_y\} = \{\min, \max\{E_{y1}, E_{y2}, E_{y3} \cdots E_{yn}\}\} \quad (9)$$

$$\{R_x, S_x\} = \{\min, \max\{v_{x1}, v_{x2}, v_{x3} \cdots v_{xm}\}\} \quad (10)$$

$$\{R_y, S_y\} = \{\min, \max\{v_{y1}, v_{y2}, v_{y3} \cdots v_{ym}\}\} \quad (11)$$

归一化处理，将 $E_{x1}$ 代入式(7)：

$$E_{x1}^* = \frac{E_{x1} - O_x}{P_x - O_x} \quad (12)$$

$$v_{y1}^* = \frac{v_{y1} - R_y}{S_x - R_y} \quad (13)$$

其余方向能耗和速度数据的归一化过程同上。

### 2.2 估价函数设计

1) 路径规划需要考虑的指标：

(1) 行走时间最短：机器人行走时间按最低的选取；(2) 能量消耗最小：机器人所走过的路径能量消耗按最低的选取。

2) 环境模型的建立：

环境建模的目的是能够为路径规划提供可用的分析平台，常用的分析建模方法有拓扑建模法和几何建模法<sup>[4,5]</sup>，在本文中选用栅格法对环境进行建模，栅格法环境建模是将仓储环境分解成一系列尺寸相同的网格单元，将环境信息或障碍物信息分布在 $l \times m \times n$ 的环境栅格矩阵中，由于每个栅格的像元值都是唯一的、确定的，所以每个栅格的标识也是唯一的、确定的<sup>[6]</sup>。建立环境模型时既要满足算法规划的准确性也要满足实时性，作以下假设：

(1) 假设搬运设备（平移、提升）无法穿越已存货位。

(2) 假设仓库内的货物实时位置已知。

(3) 假设设备性能，包括能耗、速度均已知。

(4) 假设将环境边界当作障碍物处理，设备不能越过边界行走。

假设自动化立体仓库中每一个货架长宽高尺寸分别为 $a, b, c$ ，由 $l$ 排 $m$ 列 $n$ 层的货架组成，则整个仓库有 $l \times m \times n$ 个货格，整个仓库货架区长 $a \times l$ ，宽 $b \times m$ ，高 $c \times n$ 。地图环境代价值的设置则根据仓库中已被占据的货格决定，若此货格已被占用，则无法存放或穿越，即此点为障碍点，在matlab中可建立 $l \times m \times n$ 的三维数组，其中已被占据的货格相应位置则置INF，表示障碍，未被占据的货格相应位置置0，表示可存储或可通行。

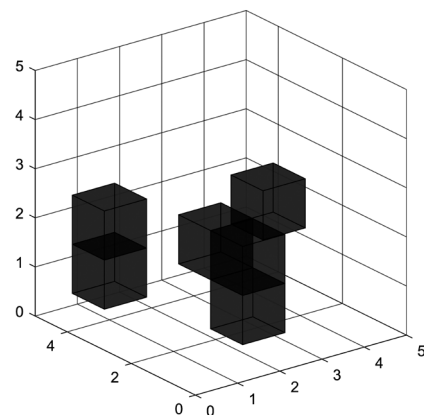


图6 仓库栅格模型示意

图6说明了一个 $4 \times 4 \times 4$ 的仓库模型，以图中填黑区域为例，简单说明本文的建模策略，则该区域的障碍障碍信息矩阵为一个 $l=m=n=4$ 的三维矩阵：

$$Z_1 = \begin{bmatrix} 0 & 0 & 0 & INF \\ INF & 0 & 0 & S \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (14)$$

$$Z_2 = \begin{bmatrix} 0 & 0 & 0 & INF \\ INF & INF & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & INF & 0 \end{bmatrix} \quad (15)$$

$$Z_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & G & 0 \end{bmatrix} \quad (16)$$

$$Z_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (17)$$

其中每层INF值所在位置即障碍点，S点代表出发点Startpoint，G点则代表终点Goalpoint。

距离函数的选取，由式(6)可知是根据曼哈顿距离扩展Z方向而来的，由于启发函数中只计算距离函数，但实际行走需要考虑拣选时间、设备能耗，且实际行走过程中拣选设备水平方向和竖直方向的运行速度和能耗并不完全一样。因此启发函数的距离计算中在竖直方向引入因子 $\gamma$ ，以调节竖直方向与水平方向的差异。

$D_{MS}$ 表示M和Startpoint点之间的距离；

$D_{MS1} = |x_s - x_m| + |y_s - y_m|$ 表示点s与下一点m水平方向之间的距离；

$D_{MS2} = \gamma |z_s - z_m|$ 表示点s与下一点m竖直方向之间的距离；

历史路径代价函数可以如下分段表达：

$$g(M) = \begin{cases} K(M') + \alpha E_x D_{MS1} + \beta \frac{D_{MS1}}{v_x} \\ K(M') + \alpha E_y D_{MS2} + \beta \frac{D_{MS2}}{v_y} \end{cases} \quad (18)$$

$K(M')$ 表示地图环境代价；

$\alpha$ 、 $\beta$ 分别表示设备能耗与运行时间在启发函数中的占比且：

$$\alpha + \beta = 1 \quad (19)$$

$\xi$ 表示启发代价调节参数（为了保证路径搜索方向始终走向目标点，仅避开障碍且不受历史路径代价函数的影响， $\xi$ 可适当调节，使得每次路径搜索都尽量靠近目标点）， $\xi$ 值的选取控制在时间和能耗参数同一量级，并略大于 $\alpha E$ 及 $\frac{\beta}{v}$ ；

$$h(M) = \xi D_{MG} \quad (20)$$

$D_{MG}$ 表示点M和路径终点G之间的距离。

$$D_{MG} = |x_m - x_g| + |y_m - y_g| + \gamma |z_m - z_g| \quad (21)$$

若 $\gamma > 1$ ，表示竖直方向单位能耗是水平方向单位能耗的 $\gamma$ 倍。则 $f(M)$ 为如下分段函数（水平、竖直）：

$$\begin{cases} g(M) + h(M) = K(M') + \alpha E_x D_{MS1} + \beta \frac{D_{MS1}}{v_x} + \xi D_{MT} \\ g(M) + h(M) = K(M') + \alpha E_y D_{MS2} + \beta \frac{D_{MS2}}{v_y} + \xi D_{MT} \end{cases} \quad (22)$$

目标函数：

$$\min \sum_S^G f(M) \quad (23)$$

## 3 仿真实验

### 3.1 算法步骤

在三维空间中求解移动机器人最优路径的方法有很多，如A\*算法、D\*算法、进化算法等<sup>[7]</sup>，在众多求解最优路径的方法中，A\*算法简单、有效，并且得到了广泛的应用，本文采用的A\*算法进行仓储设备的路径规划，其算法基本步骤描述：

Step 1: 通过InitializeField函数初始化参数，初始化地图环境；将初始节点存入SetOpen数组中。

Step2: 扩展当前的节点，即找到以当前节点为中心，其前、后、左、右、上、下6个方向邻域内的可通行节点，将不存在于SetOpen数组和SetClosed数组中的节点计算其启发代价即 $h(n)$ ，并将其存入SetOpenHeuristics数组中相对应的位置，并计算该节点的路径耗费即 $g(n)$ ，并将其设定为当前节点；在SetOpen数组中删去当前节点，并将其存入SetClosed表中。

Step3: 将扩展得到的节点依次加入SetOpen数组中，若此节点不存在SetOpen数组中，那么就将其加入到SetOpen数组中，并将计算得到的路径耗费 $g(n)$ 加入到SetOpenCost数组中相应的位置，并记录这点的来源方向到FieldPointers数组中，若此节点已经存在于SetOpen数组中，比较此点记录在SetOpenCost数组的路径消耗与新计算得到的路径消耗，若记录的路径消耗小于新计算得到的路径消耗，则对此点不做任何改变，反之，用新的路径消耗代替此点在SetOpenCost数组中相对应的位置，并更新此点的来源方向到FieldPointers数组。

Step4: SetOpenHeuristics数组中和SetOpenCost数组中相对位置的值相加, 选取其和最小的节点为当前节点, 并将其置入SetClosed数组中, 将其耗费置入SetClosedCost数组中, 然后跳转Step2。

Step5: 当SetOpen数组为空集之后, 从目标节点开始, 通过FindWayBack函数找到最终路径。

Step6: 作图并输出最终结果。

算法流程图如下:

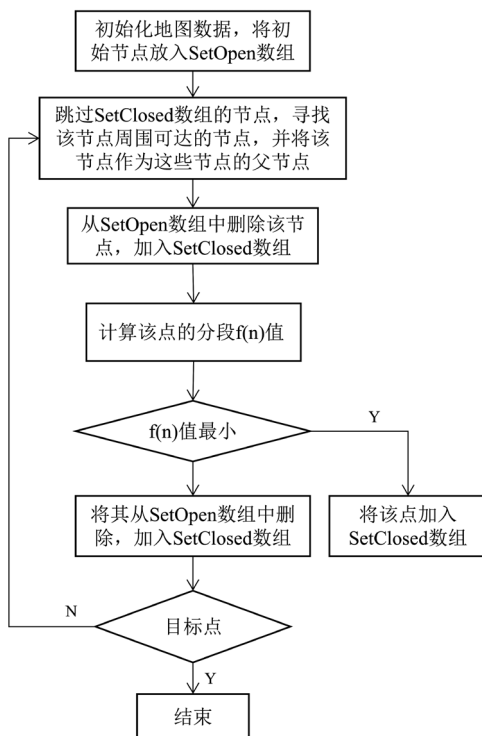


图7 算法流程图

## 3.2 算例分析

初始化数据:

假设某仓库有 $20 \times 20 \times 20$ 个库位, 合计8000个库位。其中已存货物 (即障碍点, 无法穿越)  $l=m=n=80$ ; 假设仅有一台设备, 其水平方向、竖直方向能耗、速度各异。

表1 初始化各项参数

$\alpha$	$\beta$	$\gamma$	$\xi$	$E_x$	$E_y$	$v_x$	$v_y$
0.6	0.4	2	3	1	2	2	1

初始化障碍数据: 数据量较大, 略。

初始化起点、终点: Startpoint: (6, 4, 1);

Goalpoint: (16, 15, 20)。

所得路径数据:

表2 障碍数为80时路径规划结果

X	Y	Z	X	Y	Z
16	15	20	13	9	9
15	15	20	13	9	8
14	15	20	13	9	7
14	14	20	13	9	6
14	13	20	13	8	6
14	13	19	13	8	5
13	13	19	13	7	5
13	13	18	13	6	5
13	13	17	13	6	4
13	13	16	13	6	3
13	13	15	13	5	3
13	13	14	12	5	3
12	13	14	12	4	3
12	12	14	11	4	3
12	11	14	10	4	3
12	11	13	9	4	3
13	11	13	9	4	2
13	10	13	9	4	1
13	10	12	8	4	1
13	10	11	7	4	1
13	10	10	6	4	1
13	9	10			

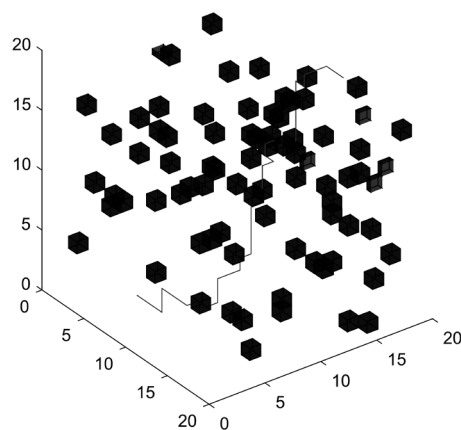


图8 80个障碍点路径规划

【下转第53页】



略,尽可能使各工位并行工作,同时适当提高气缸运动速度,缩短运动时间,进一步提高组装效率。

## 参考文献:

[1] 姬海翔,张倩,王绍宗,等.基于欧姆龙NJ控制器的门架式机器人

控制系统[J].制造业自动化,2016,38(10):7-8.

[2] 李庭贵.基于西门子PLC和变频器的物料分拣系统设计[J].制造业自动化,2013,35(6下):92-96.

[3] 廖常初.S7-200 SMART PLC应用教程[M].北京:机械工业出版社,2015.

【上接第30页】

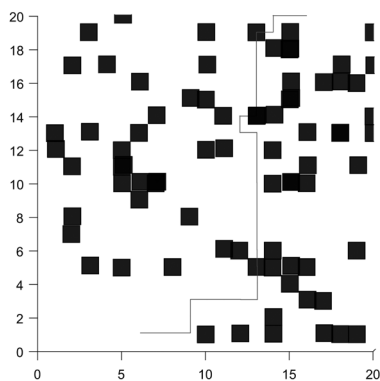


图9 左视图

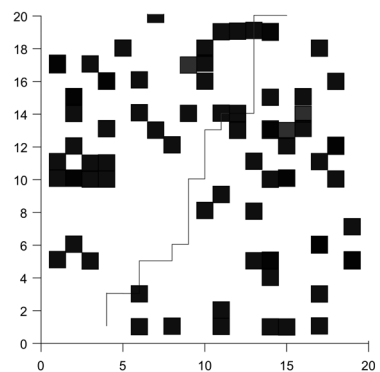


图10 主视图

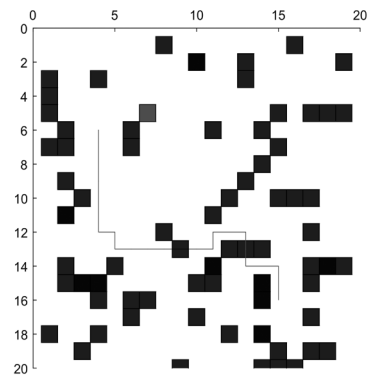


图11 俯视图

由上可知,路径规划共经过43个点,除去起点、终点,拣选路径共经过41个点,其中水平方向23个单位长度,竖直方向19个单位长度。

## 4 结论

本文通过对立体仓库环境的栅格化建模,创新性地构建了无约束环境下出入库点与仓库中任意一点的三维路径规划模型。

通过栅格化三维空间模型,将设备运行速度、设备耗能作为系统指标进行考量,针对水平方向和竖直方向的不同移动特性,提出分段路径代价函数的概念,并通过A\*算法得出该三维空间模型中的最优路径,从而得到拣选设备所有经过的路径节点。

随着仓储设备智能化的发展趋势,系统约束及边界条件越来越少,本文研究内容对于跨层作业系统、新型自动化拣选作业系统的路径规划有普适参考意义。今后可考虑多点、多设备的三维路径规划研究,对于不同仓库增加相应约束即可求解其系统的最优路径。

## 参考文献:

[1] 陈国真.多层穿梭车仓储系统配置优化[D].山东大学,2016.

[2] 陈洋,赵新刚,韩建达.移动机器人3维路径规划方法综述[J].机器人,2010,32(4):570-571.

[3] Ren Tianzhu,Zhou Rui,XiaJie,Dong Zhuoning.Three-dimensional Path Planning of UAV Based On an Improved A\* Algorithm[A].Proceedings of 2016 IEEE Chinese Guidance,Navigation and Control Conference[C],August 12-14,2016:141-142.

[4] Boschian V,Pruski A.Grid Modeling of robot cells:A memory-efficient approach[J].Journal of Intelligent and Robotic Systems,1993,8(2):201-223.

[5] Alami R,Fleury S,Herrb M,et al.Multi-robot cooperation in the MARTHA project[J].Robotics&Automation Magazine,IEEE,1998,5(1):36-47.

[6] 潘思宇,徐向荣.基于A\*算法的机器人二维与三维路径规划实现[J].井冈山大学学报,2015,36(5):85-87.

[7] 陆亮,王佳琪,宗成星,赵萍.基于A\*算法的四轴飞行器三维路径规划仿真[J].合肥工业大学学报,2017,40(3):306-307.