

## ◎工程与应用◎

## 对AGV路径规划A星算法的改进与验证

赵江, 张岩, 马泽文, 叶子超

ZHAO Jiang, ZHANG Yan, MA Zewen, YE Zichao

河北科技大学 电气工程学院, 石家庄 050018

College of Electrical Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China

ZHAO Jiang, ZHANG Yan, MA Zewen, et al. Improvement and verification of A-star algorithm for AGV path planning. *Computer Engineering and Applications*, 2018, 54(21): 217-223.

**Abstract:** To reduce the length of the path and the number of turns of the Automate Guide Vehicle (AGV), an improved A-star algorithm is proposed. The path planned by the traditional A-star algorithm is further optimized by geometric methods. Firstly, based on the initial path planned by A-star algorithm, traversing all the nodes on the initial path, removing redundant inflection points and redundant nodes and obtaining the path that contains the starting node, the end node, and the key inflection points. Then the rotation direction and rotation angle of AGV at the inflection point are calculated to adjust the posture of the AGV. Comparative experiments on traditional A-star algorithm, ant colony algorithm and improved A-star algorithm are carried out respectively. And the experimental results demonstrate that the method not only has a fast calculation speed but also can provide a short and smooth path.

**Key words:** Automate Guide Vehicle (AGV); A-star algorithm; path planning; heuristic search algorithm

**摘要:** 为了减少AGV (Automate Guide Vehicle, 自动导引车) 的运输路径长度和转折次数, 提出了改进的A星算法, 采用几何方法对传统A星算法规划出的路径进行进一步优化。首先遍历路径上的所有节点, 剔除路径中冗余节点和不必要拐点, 获取仅包含起点、必要拐点、终点的路径。最后计算AGV在拐点处的旋转角度及旋转方向, 使AGV在拐点处能够调整自身姿态。并分别对传统A星算法、蚁群算法和改进A星算法进行了对比实验。实验结果表明该方法不仅保留了A星算法运算速度快的优点, 还能够有效地规划出距离短且平滑的路径。提高了AGV的运行效率, 降低了AGV的耗能。

**关键词:** 自动导引车; A星算法; 路径规划; 启发式搜索算

**文献标志码:** A **中图分类号:** TP18 **doi:** 10.3778/j.issn.1002-8331.1806-0069

## 1 引言

近几年来, AGV (自动导引车) 已广泛应用于仓储、物流等领域<sup>[1-2]</sup>。对于AGV而言, 尤其是港口、码头中的大型AGV, 路径的选择对AGV的运输效率影响很大<sup>[3]</sup>。过长的运输路径使AGV的耗能大大增加, 增加运输成本; 大型AGV转弯半径大, 频繁的转弯不仅会使AGV耗能增加, 同时由于惯性的影响, 也不利于大型

AGV的安全。所以AGV的路径规划是AGV系统研究的关键点之一。

路径规划<sup>[4]</sup>是指在周围环境存在障碍物的情况下, 为控制对象<sup>[5]</sup>规划一条从起点到终点的 safely 的最优路径<sup>[6-7]</sup>。目前应用较多的路径规划算法有遗传算法、蚁群算法、A星算法等<sup>[8-9]</sup>。遗传算法具有良好的全局优化能力, 但其收敛速度较慢, 运算效率不高<sup>[10-11]</sup>。文献[11]设

**基金项目:** 河北省高等学校科学技术研究项目 (No.ZD2016142); 河北省引进国外智力项目 (No.1200343); 国家自然科学基金 (No.51507048); 河北省大学生创新创业训练项目; 河北科技大学通用航空平台项目。

**作者简介:** 赵江 (1958—), 男, 教授, 硕士研究生导师, 主要研究领域为过程参数检测与智能控制技术; 张岩 (1983—), 男, 博士, 讲师, 主要研究领域为智能控制技术; 马泽文 (1993—), 男, 硕士研究生, 主要研究领域为地面机器人避障及定位技术, E-mail: 601628752@qq.com; 叶子超 (1995—), 男, 本科生, 主要研究领域为智能控制技术。

**收稿日期:** 2018-06-08 **修回日期:** 2018-08-06 **文章编号:** 1002-8331(2018)21-0217-07

计了一种知识引导遗传算法,在染色体的编码、各种遗传算子和优化算子中加入了相关的领域知识,提高了收敛速度和求解效率,但该方法容易陷入局部最优解。蚁群算法求解的平均路径最短,但其运行时间较长<sup>[12-13]</sup>。文献[14]在蚁群算法搜索过程中融合了人工势场算法,将人工势场法中力因素转换为局部扩散信息素,提高了蚁群算法的收敛速度,但计算复杂度较大。A星算法因其快速的计算速度广泛应用于AGV的路径规划问题,但其规划的路径上存在冗余节点和不必要拐点较多等问题<sup>[15-16]</sup>。王殿军根据移动机器人的实际运行环境采用栅格法构建了环境地图并提出了一种改进A星算法,该方法剔除了路径的冗余节点,但无法剔除路中的多余拐点<sup>[17]</sup>。Korf等人在A星算法的基础上融合了迭代加深算法,无需进行估价排序,优化了A星算法的寻路策略<sup>[18]</sup>。文献[19]将传统A算法的可搜索邻域个数从8个拓展为无限个。该方法可以向任意方向搜索最短路径,但搜索节点过多导致算法运算时间较长。文献[20]提出了HPA\*算法,通过减小搜索空间来提高搜索速度,提高了寻路效率,但它往往得不到较好的路径。

以上方法有的可以规划出良好的路径但计算时间较长,有的缩短了计算时间但无法规划出良好的路径。基于以上问题提出一种改进的A星算法,在传统A星算法所规划出的路径的基础上,剔除路径中的冗余节点和多余拐点,获得仅包含起点、必要拐点、终点的新路径,并计算出了AGV在新路径中拐点出的旋转角度和旋转方向,使AGV在拐点处能够调整姿态。不仅保留了A星算法运算速度快的优点,还能够消除路径中的不必要拐点和冗余节点。本文分别将A星算法、蚁群算法和改进后的A星算法移植到了实际的AGV系统上进行了实验研究,并且通过实验得出:改进的后A星算法所规划出路径的拐点要少于A星算法与蚁群算法所规划出路径的拐点,改进的后A星算法所规划出路径长度要少于A星算法所规划出路径长度与蚁群算法所规划出路径长度相差不多,但运算时间要短于蚁群算法。改进后的A星算法提高了AGV路径的平滑性,缩短了路径的长度还节省了AGV路径规划的时间,适用于港口、码头运输中大型AGV的实时路径规划。

## 2 A星算法

A星算法是一种启发式算法,其路径长度评价函数 $F(n)$ 是用来估算路径总长度的函数,其计算公式如下:

$$F(n) = G(n) + H(n) \quad (1)$$

式(1)中, $G(n)$ 为沿着所规划出的路径从起点到当前节点 $n$ 所花费的实际代价。 $H(n)$ 为启发函数,用来估算当前节点 $n$ 到终点花费的预估代价,其计算公式如下:

$$H(n) = \text{sqrt}[(x_n - x_e)^2 + (y_n - y_e)^2] \quad (2)$$

式(2)中, $(x_n, y_n)$ 为当前节点 $n$ 的坐标, $(x_e, y_e)$ 为终点坐标。

A星算法在搜索路径过程中需要建立两个表格:open表和close表,open表用于存放当前节点(父节点)周围八个方向的节点(子节点)及子节点的 $F(n)$ 值,并且表中的节点需要根据 $F(n)$ 值的大小按从小到大的顺序排序。close表用于存放规划路径上的节点,close表中节点所组成的路径即为规划出的路径。A星算法主要以路径长度评价函数 $F(n)$ 作为评价指标,主要流程为:在平面坐标系内,从起点开始,将起点放入close表中,并将起点作为当前节点(父节点),对父节点周围八个子节点分别计算 $F(n)$ 值。并将子节点及子节点的 $F(n)$ 值放入open表中排序,找出 $F(n)$ 值最小的子节点将该节点放入close表中,并将该节点作为新的当前节点(父节点)。并如此循环,直到找到终点。其中close表中的节点所组成的路径就是A星算法所规划的路径。

## 3 改进后的A星算法

提出了一种改进的A星算法,采用几何方法对传统A星算法规划出的路径进行进一步优化,得到长度更短,拐点更少的路径。改进后的A星算法主要包括三部分:(1)合并路径上的共线节点。(2)消除路径上的多余拐点。(3)计算AGV在优化后路径拐点出的旋转方向和旋转角度。

### 3.1 消除共线节点的方法

设节点B为路径中的某一节点,坐标为 $(x_2, y_2)$ ,节点A为节点B的前一节点,坐标为 $(x_1, y_1)$ ,节点C为节点B的后一节点坐标为 $(x_3, y_3)$ ,如果A、B所在直线的斜率与B、C所在直线的斜率相同则可以判定三点共线:

$$S_{AB} = \frac{y_2 - y_1}{x_2 - x_1} \quad (3)$$

$$S_{BC} = \frac{y_3 - y_2}{x_3 - x_2} \quad (4)$$

如上所示, $S_{AB}$ 为线段AB的斜率, $S_{BC}$ 为线段BC的斜率,如果 $S_{AB} = S_{BC}$ 则A、B、C三点共线,节点B为多余节点,剔除节点B,路径由A-B-C变为A-C。

### 3.2 消除多余拐点的方法

点位置的判断:如图1所示,平面内存在四个点分别为 $Q_0, Q_1, Q_2, Q_3$ ,设 $Q_0$ 为原点, $Q_1, Q_2, Q_3$ 坐标分别为 $(x_4, y_4), (x_5, y_5), (x_6, y_6)$ ,由 $Q_0$ 指向 $Q_1$ 的向量为 $n$ ,由 $Q_0$ 指向 $Q_2$ 的向量为 $p$ ,由 $Q_0$ 指向 $Q_3$ 的向量为 $q$ ,可以把 $n, p, q$ 看成 $Z$ 轴值恒为0的三维向量,坐标分别为 $(x_4, y_4, 0), (x_5, y_5, 0), (x_6, y_6, 0)$ , $n$ 与 $p$ 的叉积以及 $n$ 与 $q$ 的叉积分别为:

$$n \times p = \begin{vmatrix} i & j & k \\ x_4 & y_4 & 0 \\ x_5 & y_5 & 0 \end{vmatrix} = (x_4 \cdot y_5 - x_5 \cdot y_4)k = ak \quad (5)$$

$$n \times q = \begin{vmatrix} i & j & k \\ x_4 & y_4 & 0 \\ x_6 & y_6 & 0 \end{vmatrix} = (x_4 \cdot y_6 - x_6 \cdot y_4)k = bk \quad (6)$$

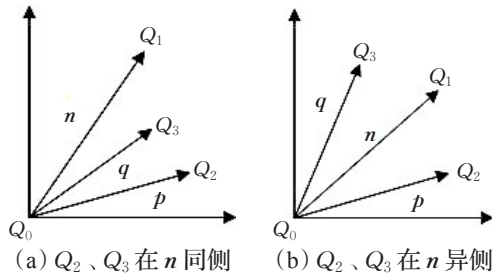


图1 两点位置的判断

规定垂直于纸面向外的方向为正方向,其中公式(5)、(6)中的 \$a\$ 和 \$b\$ 为叉积的系数。根据右手定则可知 \$n\$ 与 \$p\$ 的叉积垂直于纸面向里,即 \$a < 0\$, \$p\$ 在 \$n\$ 的顺时针方向, \$Q\_2\$ 在 \$n\$ 的顺时针方向, \$n\$ 与 \$q\$ 的叉积垂直于纸面向外,即 \$b > 0\$, \$q\$ 在 \$n\$ 的逆时针方向, \$Q\_3\$ 在 \$n\$ 的逆时针方向,由此可得:当 \$a \cdot b > 0\$, \$Q\_2\$、\$Q\_3\$ 位于 \$n\$ 同侧,如图 1(a)所示,当 \$a \cdot b < 0\$, \$Q\_2\$、\$Q\_3\$ 位于 \$n\$ 异侧,如图 1(b)所示。

线段相交的判定:设平面内存在四个点, \$P\_1\$、\$P\_2\$、\$P\_3\$、\$P\_4\$ 组成线段 \$P\_1P\_2\$ 和线段 \$P\_3P\_4\$, 如果 \$P\_1\$、\$P\_2\$ 位于线段 \$P\_3P\_4\$ 的两侧且 \$P\_3\$、\$P\_4\$ 位于线段 \$P\_1P\_2\$ 的两侧,则可以判定两个线段相交。如图 2 所示,设由 \$P\_3\$ 指向 \$P\_4\$ 的向量为 \$q\$, 由 \$P\_3\$ 指向 \$P\_2\$ 的向量为 \$p\$, 由 \$P\_3\$ 指向 \$P\_1\$ 的向量为 \$l\$, 由 \$P\_1\$ 指向 \$P\_3\$ 的向量为 \$r\$, 由 \$P\_1\$ 指向 \$P\_2\$ 的向量为 \$n\$, 由 \$P\_1\$ 指向 \$P\_4\$ 的向量为 \$m\$, 求叉积可得:

$$q \times p = c_1 k \quad (7)$$

$$q \times l = c_2 k \quad (8)$$

$$n \times m = c_3 k \quad (9)$$

$$n \times r = c_4 k \quad (10)$$

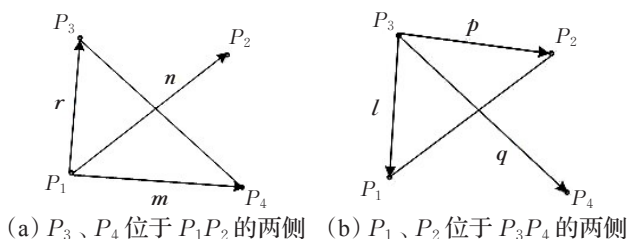


图2 线段相交的判定

公式(7)~(10)中的 \$c\_1\$、\$c\_2\$、\$c\_3\$、\$c\_4\$ 均为叉积的系数。综上所述,当 \$c\_1 \cdot c\_2 < 0\$ 且 \$c\_3 \cdot c\_4 < 0\$ 时,线段 \$P\_1P\_2\$ 和线段 \$P\_3P\_4\$ 相交。

剔除多余节点的方法:连接路径中某一节点的前一节点与后一节点,如果该点的前一节点与后一节点所连接的线段上不存在障碍物,则剔除该点。

是否存在障碍物的判定:判断前一节点与后一节点所连接的线段上是否存在障碍物的方法就是判断该线段是否与障碍物其中的一条对角线相交,如果相交则说明有障碍物,如果不相交则说明没有障碍物。假设节点 \$B\$ 为路径上的某一节点,坐标为 \$(x\_2, y\_2)\$, 节点 \$A\$ 为节点 \$B\$ 的前一节点,坐标为 \$(x\_1, y\_1)\$, 节点 \$C\$ 为节点 \$B\$ 的后一节点坐标为 \$(x\_3, y\_3)\$。点 \$P\$、\$Q\$、\$N\$、\$M\$ 为障碍物的四个顶点,坐标分别为 \$(x\_7, y\_7)\$、\$(x\_8, y\_8)\$、\$(x\_9, y\_9)\$、\$(x\_{10}, y\_{10})\$。设由 \$A\$ 指向 \$C\$ 的向量为 \$a\$, 由 \$A\$ 指向 \$Q\$ 的向量为 \$b\$, 由 \$A\$ 指向 \$N\$ 的向量为 \$c\$, 由 \$N\$ 指向 \$Q\$ 的向量为 \$d\$, 由 \$N\$ 指向 \$A\$ 之间的向量为 \$e\$, 由 \$N\$ 指向 \$C\$ 的向量为 \$f\$:

$$a \times b = [(x_1 - x_3)(y_1 - y_8) - (y_1 - y_3)(x_1 - x_8)]k = w_1 k \quad (11)$$

$$a \times c = [(x_1 - x_3)(y_1 - y_9) - (y_1 - y_3)(x_1 - x_9)]k = w_2 k \quad (12)$$

$$d \times e = [(x_9 - x_8)(y_9 - y_1) - (y_9 - y_8)(x_9 - x_1)]k = w_3 k \quad (13)$$

$$d \times f = [(x_9 - x_8)(y_9 - y_3) - (y_9 - y_8)(x_9 - x_3)]k = w_4 k \quad (14)$$

公式(11)~(14)中的 \$w\_1\$、\$w\_2\$、\$w\_3\$、\$w\_4\$ 均为叉积的系数。如果 \$w\_1 \cdot w\_2 < 0\$ 且 \$w\_3 \cdot w\_4 < 0\$ 则说明线段 \$AC\$ 与对角线 \$NQ\$ 相交如图 3(a)所示反之则不相交如图 3(b)所示。同理,也可判断对角线 \$PM\$ 与线段 \$AC\$ 是否相交。线段 \$AC\$ 只要与其中一条对角线相交,则说明节点 \$A\$ 与节点 \$C\$ 之间存在障碍物,原路径不变,依然为 \$A\$-\$B\$-\$C\$, 如果线段 \$AC\$ 与两条对角线都没有相交,则说明节点 \$A\$ 与节点 \$C\$ 之间不存在障碍物,节点 \$B\$ 为多余拐点,连接节点 \$A\$ 与节点 \$C\$, 剔除节点 \$B\$, 路径由 \$A\$-\$B\$-\$C\$ 变为 \$A\$-\$C\$。

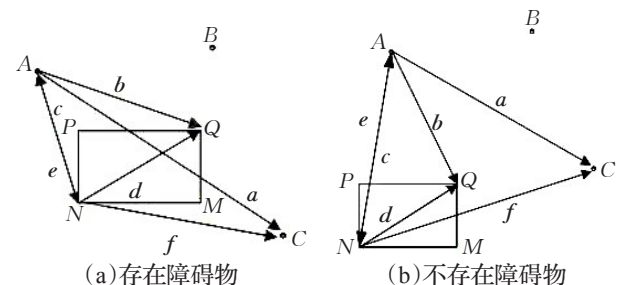


图3 是否存在障碍物的判定

改进后的 A 星算法流程:该流程为获得 A 星算法规划的初始路径后的流程。如图 4 所示: \$P\_i\$ 为 A 星算法规划的初始路径的节点, \$i\$ 为原始路径节点的标号, 节点 \$P\_i\$ 为路径中的某一节点, 节点 \$P\_{i+1}\$ 为 \$P\_i\$ 的下一节点, 节点 \$P\_{i+2}\$ 为 \$P\_i\$ 的下一节点。

步骤 1 取初始路径的第一个节点,使 \$i=1\$。

步骤 2 判断节点 \$P\_{i+1}\$ 是否为终点,如果不是终点,运行步骤 3,如果是终点,结束算法,初始路径中剩余节点所组成的路径为新规划的路径。

步骤 3 运用 2.1 节介绍的方法判断节点 \$P\_i\$、\$P\_{i+1}\$、\$P\_{i+2}\$ 是否共线,如果不共线,运行步骤 4,如果共线,删除接节点 \$P\_{i+1}\$ 并且返回步骤 2。因为节点 \$P\_{i+1}\$ 被删除了,所以节点 \$P\_i\$ 后面的点的标号自动减 1。

步骤 4 运用 2.2 节介绍的方法判断节点 \$P\_i\$ 和节点



$P_{i+2}$  之间是否存在障碍物, 如果不存在障碍物, 删除节点  $P_{i+1}$  并且返回步骤2, 如果存在障碍物, 运行步骤5。

步骤5 令  $i=i+1$ , 并返回步骤2, 进入下一个节点的判断。

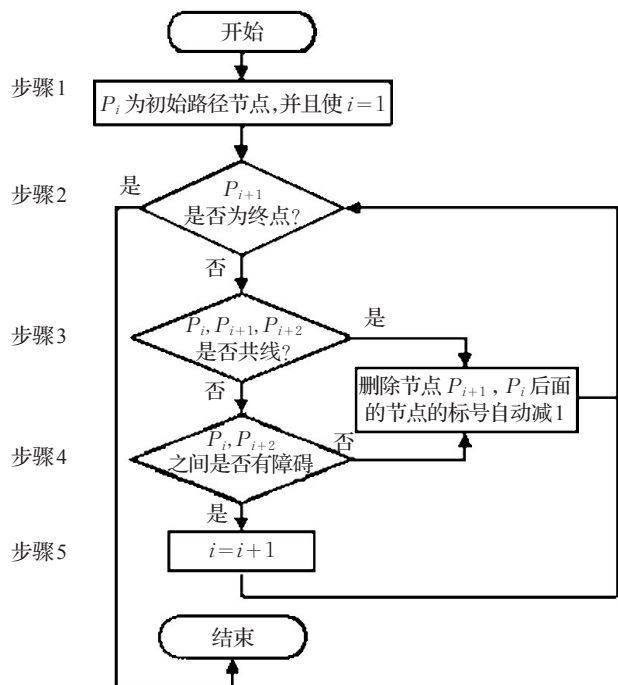


图4 改进后的A星算法流程图

### 3.3 计算改进路径拐点处AGV旋转角度及方向

设路径上有三个点, 点  $A$ 、 $B$ 、 $C$  坐标分别为  $(x_1, y_1)$ 、 $(x_2, y_2)$ 、 $(x_3, y_3)$ , 如图5所示, 由  $A$  指向  $B$  的向量为  $AB$ , 由  $B$  指向  $C$  的向量为  $BC$ , 向量  $AB$  与  $x$  轴夹角为  $\alpha_1$ , 向量  $BC$  与  $x$  轴夹角为  $\alpha_2$ , 将向量  $AB$  平移, 使点  $A$  与点  $B$  重合, 得到向量  $BD$ , 向量  $BD$  与向量  $BC$  的夹角为  $\theta$ , 点  $D$  到点  $C$  的向量为  $DC$ 。

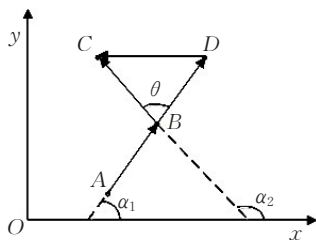


图5 路径的向量表示

向量可表示为:

$$AB = BD = (x_2 - x_1)i + (y_2 - y_1)j \quad (15)$$

$$BC = (x_3 - x_2)i + (y_3 - y_2)j \quad (16)$$

$$\cos \theta = (BD \cdot BC) / \sqrt{|BD|} \sqrt{|BC|} = \frac{(x_3 - x_2)(x_2 - x_1) + (y_3 - y_2)(y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}} \quad (17)$$

令  $\beta = \alpha_1 - \alpha_2$ , 当  $\beta > 0$  且  $y_3 \geq y_2$  且  $y_2 \geq y_1$  时 AGV 的旋转角度为  $\theta$ , 旋转方向为  $DC$  的方向。

基于上述方法可以计算出各种情况下 AGV 的旋转

角度和旋转方向, 结果如下:

情况1

$$\{\beta > 0 | \beta \in R\} \cap \{y_3 \geq y_2 | y_3, y_2 \in R\} \cap \{y_2 \geq y_1 | y_2, y_1 \in R\}$$

AGV 逆时针旋转  $\theta$ ;

情况2

$$\{\beta > 0 | \beta \in R\} \cap \{y_3 \geq y_2 | y_3, y_2 \in R\} \cap \{y_2 < y_1 | y_2, y_1 \in R\}$$

AGV 顺时针旋转  $\theta$ ;

情况3

$$\{\beta > 0 | \beta \in R\} \cap \{y_3 < y_2 | y_3, y_2 \in R\} \cap \{y_2 \geq y_1 | y_2, y_1 \in R\}$$

AGV 顺时针旋转  $\theta$ ;

情况4

$$\{\beta > 0 | \beta \in R\} \cap \{y_3 < y_2 | y_3, y_2 \in R\} \cap \{y_2 < y_1 | y_2, y_1 \in R\}$$

AGV 逆时针旋转  $\theta$ ;

情况5

$$\{\beta < 0 | \beta \in R\} \cap \{y_3 \geq y_2 | y_3, y_2 \in R\} \cap \{y_2 \geq y_1 | y_2, y_1 \in R\}$$

AGV 顺时针旋转  $\theta$ ;

情况6

$$\{\beta < 0 | \beta \in R\} \cap \{y_3 \geq y_2 | y_3, y_2 \in R\} \cap \{y_2 < y_1 | y_2, y_1 \in R\}$$

AGV 逆时针旋转  $\theta$ ;

情况7

$$\{\beta < 0 | \beta \in R\} \cap \{y_3 < y_2 | y_3, y_2 \in R\} \cap \{y_2 \geq y_1 | y_2, y_1 \in R\}$$

AGV 逆时针旋转  $\theta$ ;

情况8

$$\{\beta < 0 | \beta \in R\} \cap \{y_3 < y_2 | y_3, y_2 \in R\} \cap \{y_2 < y_1 | y_2, y_1 \in R\}$$

AGV 顺时针旋转  $\theta$ 。

## 4 实验分析

本实验采用自主开发的 AGV 系统, 该系统包括 AGV 移动平台、上位机系统, 以及超声波定位接收装置。其中 AGV 移动平台包括无线通讯接收模块、上位机通讯模块、主控单片机、超声波发射模块、激光扫描仪、液晶显示屏。定位装置固定在屋顶, 包括超声波接收模块, 无线通讯模块, 主控单片机。上位机系统是以 Labview 为开发环境, 并在 Win7 系统下运行, 上位机系统主要包括参数设置模块, 算法处理模块电子地图, 运行轨迹显示模块以及下位机通讯模块。

该系统的主要框图如图6所示, 图7为实际的 AGV 系统。

首先上位机系统的参数设置模块设定电子地图的大小、障碍物的排布和起点终点的坐标, 算法处理模块根据设定的参数计算得到电子地图与路径轨迹, 并在电子地图及运行轨迹显示模块中显示。下位机通讯模块将电子地图以及路径轨迹的节点信息发送到 AGV 移动平台的上位机通讯模块, AGV 移动平台的主控单片机根据收到路径信息控制 AGV 运动, 同时控制超声波发射模块发射超声波, 并通过无线通讯模块向超声波定位

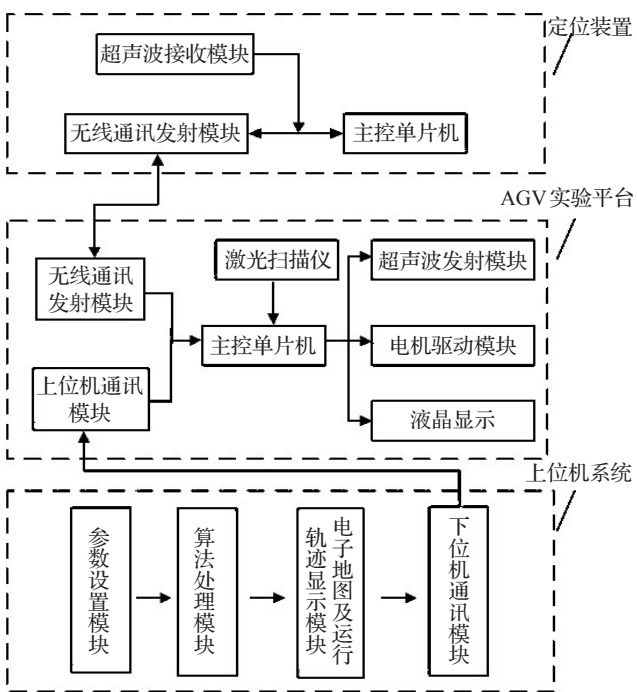


图6 AGV系统框图

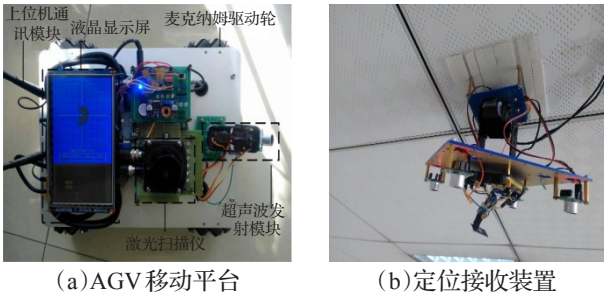
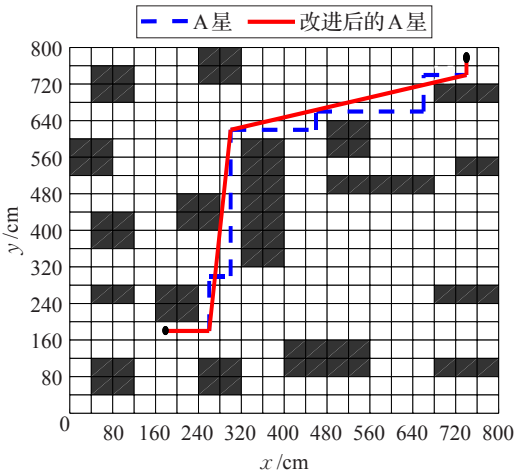


图7 AGV系统

接受装置发送命令,令超声波定位接受装置开启定时器功能。超声波定位接受装置根据接收到超声波的时间结合超声波的传播速度可得到超声波发射模块到超声波接收模块的距离,再根据特定的几何算法计算得到AGV移动平台的坐标,并通过无线通讯模块将坐标传送到AGV移动平台,在液晶显示屏上完成坐标显示。



(a)起点为(180,180)终点为(740,780)路径

本文在800 cm×800 cm的环境区域内对改进的A星算法进行验证,首先将800 cm×800 cm的环境区域转换为20×20的栅格地图,如图8所示,其中黑色栅格代表障碍物,白色栅格代表可通行的区域,图9为在800 cm×800 cm的环境区域运行的AGV,其中箱子代表障碍物。

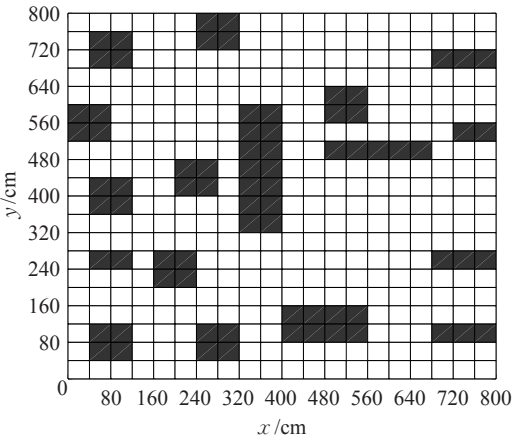


图8 800 cm×800 cm环境的栅格地图

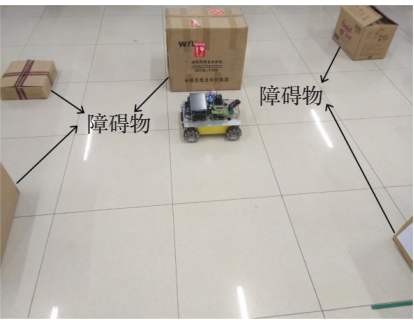
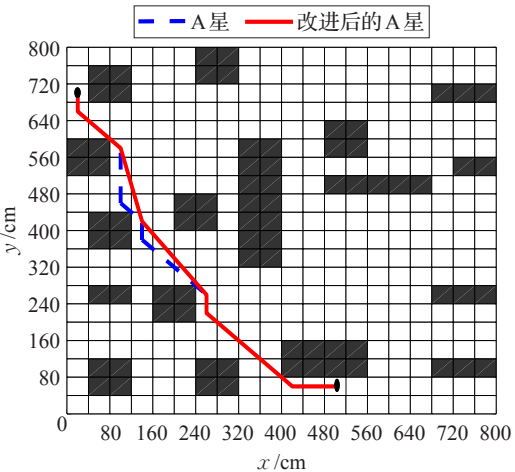


图9 800 cm×800 cm环境下运行的AGV

A星算法与改进后的A星算法运行轨迹对比图如图10所示,其中图10(a)为在起点为(180,180)终点为(740,780)的情况下两算法规划出的路径轨迹,图10(b)为在起点为(500,60)终点为(20,700)的情况下两算法规划出的路径轨迹。表1为AGV在800 cm×800 cm环境下运行的实验数据。



(b)起点为(500,60)终点为(20,700)的路径

图10 A星算法与改进后A星算法规划出的路径轨迹

由表1数据可知,在起点为(180,180),终点为(740,780)的情况下,改进后的A星算法比A星算法路径长度缩短了约137 cm,长度缩短了12%左右,拐点减少了6次,减少了66%。在起点为(500,60)终点为(20,700)的情况下,改进后的A星算法比A星算法路径长度缩短了约45 cm,长度缩短了5%左右,拐点减少了2次,减少了25%。在起点为(220,180)终点为(620,700)的情况下,改进后的A星算法比A星算法路径长度缩短了约140 cm,长度缩短了17.6%左右,拐点减少了3次,减少了50%。

为了进一步验证改进后的A星算法的有效性和可行性,本文对改进后的A星算法与蚁群算法进行了对比,蚁群算法与改进后的A星算法运行轨迹对比图如图11所示,其中图11(a)为在起点为(180,180)终点为(740,780)的情况下两算法规划出的路径轨迹,图11(b)为在起点为(500,60)终点为(20,700)的情况下两算法规划出的路径轨迹。表2为AGV在800 cm×800 cm环境下运行的实验数据。

由表2数据可知在起点为(180,180)终点为(740,780)的情况下,改进后的A星算法比蚁群算法路径长度增加了约75 cm,长度增加了8%左右,拐点减少了8次,减少了70%,运算时间缩短了93%。在起点为(500,60)终点为(20,700)的情况下,改进后的A星算法比蚁群算

法路径长度减少了约41 cm,长度缩短了4.6%左右,拐点减少了4次,减少了40%,运算时间缩短了95%。在起点为(220,180)终点为(620,700)的情况下,改进后的A星算法比蚁群算法路径长度减少了约44 cm,长度缩短了6.3%左右,拐点减少了2次,减少了40%,运算时间缩短了95%。

本文在800 cm×800 cm区域内,起点为(180,180),终点为(740,780),障碍物密度为15%、20%、25%、30%、35%、40%、45%、50%、55%情况下的进行了实验如图12所示。

由图12可以看出,由于障碍物密度的增加,算法的多余拐点和冗余节点会减少,所以路径长度减少量和转折次数减少量也会随之降低。如图12(a)所示随着障碍物密度的增加,改进后的A星算法所规划出的路径的长度相比于A星算法所规划出路径的长度减少了5%~30%,与蚁群算法所规划出路径的长度相差不多。如图12(b)随着障碍物密度的增加,改进后的A星算法所规划出路径的转折次数比A星算法的所规划出路径的转折次数减少了25%~70%,比蚁群算法所规划出的路径的转折次数减少了30%~70%。

## 5 结论

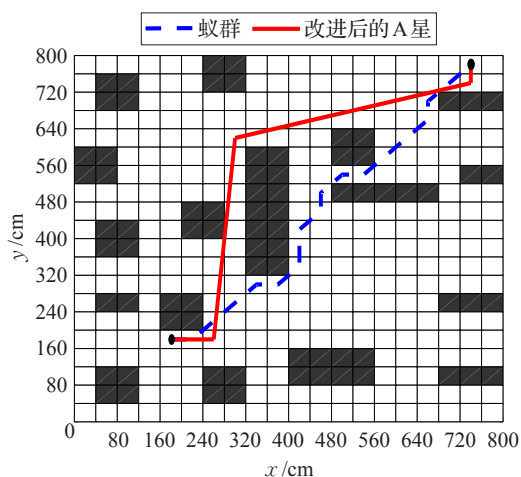
本文对A星算法进行了改进,改进后的A星算法能够有效减少AGV的运输路径长度和转折次数,并且具

表1 A星与改进后A星在800 cm×800 cm的区域内运行的实验数据

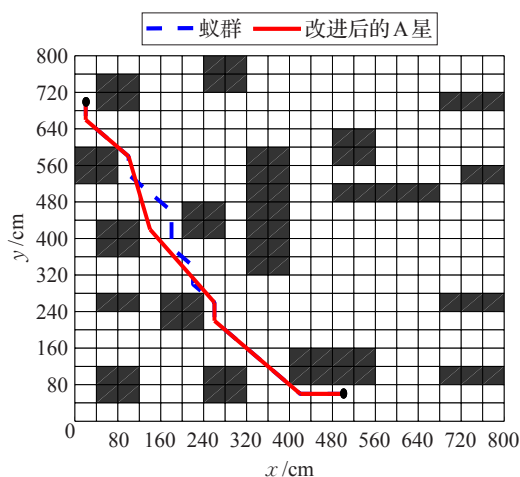
起点 终点/cm	算法	路径长度/cm	转折次数
(180,180)	A星	1 141	9
(740,780)	改进A星	1 004	3
(500,60)	A星	890	8
(20,700)	改进A星	845	6
(220,180)	A星	795	6
(620,700)	改进A星	655	3

表2 改进A星算法与蚁群算法在800 cm×800 cm的区域内运行的实验数据

起点 终点/cm	算法	路径 长度/cm	转折次数	运算 时间/s
(180,180)	蚁群	942	11	7.847
(740,780)	改进A星	1 004	3	0.549
(500,60)	蚁群	886	10	7.362
(20,700)	改进A星	845	6	0.186
(220,180)	蚁群	699	5	7.526
(620,700)	改进A星	655	3	0.328



(a)起点为(180,180)终点为(740,780)路径



(b)起点为(500,60)终点为(20,700)的路径

图11 蚁群算法与改进A星算法规划出的路径轨迹

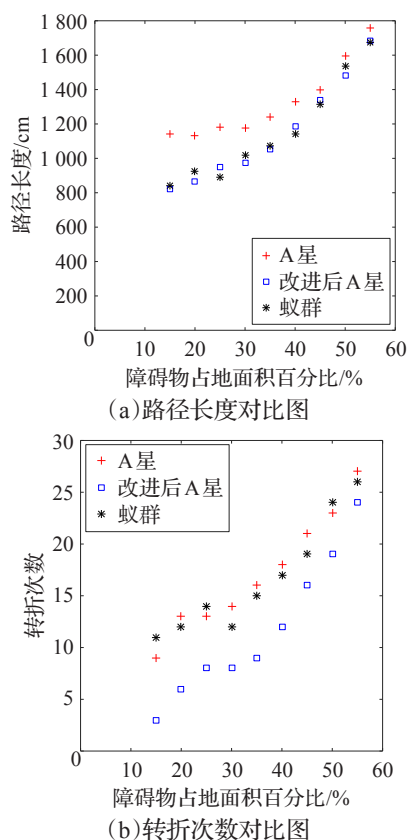


图12 不同障碍物密度下的实验数据

有较快的运算速度。该方法通过比较路径中某一节点与其前一节点之间的线段的斜率和该点与其后一节点之间的线段的斜率的方法来消除路径中的冗余节点。通过判断当路径中某一节点的前一节点与后一节点之间的连线上是否存在障碍物的方法来消除路径中的多余拐点。改进后的A星算法相比于A星算法路径长度减少了2%~25%,转折次数减少了20%~70%。相比于蚁群算法路径长度相差不多,转折次数减少了30%~70%,运算速度快了95%。本方法不仅保留了A星算法运算速度快的优点,还具有较少的拐点,和较短的路径长度,为提高大型AGV的运行效率提供了新思路。

### 参考文献:

- [1] 钟建琳, Maslowski A. 制造环境中AGV运输子系统的路径规划[J]. 机械设计与制造, 2010(2): 237-239.
- [2] Rafael D. Algorithm for path planning, navigation and guidance of an AGV[J]. Robotics and Autonomous System, 1991, 7(4): 309-326.
- [3] Li Bai, Liu Hong, Duo Xiao, et al. Centralize and optimal motion planning for large-scale AGV systems: a generic approach[J]. Advance in Engineering Software, 2017, 106: 33-46.
- [4] Han J, Seo Y. Mobile robot path planning with surrounding point set and path improvement[J]. Applied Soft Computing, 2017, 57: 35-47.
- [5] Mac T T, Copot C, Tran D T, et al. Heuristic approaches in robot path planning: a survey[J]. Robotics and Autonomous System, 2016, 86: 13-28.
- [6] Wang An, Xu Jing, Chen Heping, et al. Optimal path planning and control of assemble robots for hard-measuring easy-deformation assemblies[J]. IEEE/ASME Transactions on Mechatronics, 2017, 22(4): 1600-1609.
- [7] Guruj A K, Agarwal H, Parsediya D K. Time-efficient A\* algorithm for robot path planning[J]. Procedia Technology, 2016, 23: 144-149.
- [8] Rashid A T, Ali A A, Frasca M, et al. Path planning with obstacle avoidance based on visibility binary tree algorithm[J]. Robotics and Autonomous System, 2013, 61(12): 1440-1449.
- [9] 朱大奇, 颜明重. 移动机器人路径规划技术综述[J]. 控制与决策, 2010, 25(7): 961-967.
- [10] 雷伟军, 程筱胜, 戴宁, 等. 基于改进遗传算法的多模型加工路径规划[J]. 机械工程学报, 2014, 50(11): 153-161.
- [11] 王雪松, 高阳, 程玉虎, 等. 知识引导遗传算法实现机器人路径规划[J]. 控制与决策, 2009, 24: 1043-1049.
- [12] 侯梦婷, 赵作鹏, 高萌, 等. 采用角度因子的蚁群优化多路径路由算法[J]. 计算机工程与应用, 2017, 53(1): 107-112.
- [13] 万正宜, 彭玉旭. 求解旅行商问题的改进型量子蚁群算法[J]. 计算机工程与应用, 2016, 52(22): 59-63.
- [14] 刘建华, 杨建国, 刘华平, 等. 基于势场蚁群算法的移动机器人全局路径规划方法[J]. 农业机械学报, 2015, 46(9): 18-27.
- [15] 王卫红. 基于改进A星算法的移动机器人路径规划[J]. 同济大学学报(自然科学版), 2010, 38(11): 1647-1651.
- [16] 程传奇, 郝向阳, 李建胜, 等. 融合改进A\*算法和动态窗口法的全局动态路径规划[J]. 西安交通大学学报, 2017, 51(11): 137-143.
- [17] 王殿军. 基于改进A星算法的室内移动机器人路径规划[J]. 清华大学学报(自然科学版), 2012, 52(8): 1085-1089.
- [18] Korf R E. Depth-first iterative-deepening: an optimal admissible tree search[J]. Artificial Intelligence, 1985, 27(1): 97-109.
- [19] 辛煜, 梁华为, 杜明博, 等. 一种可搜索无限个邻域的改进A\*算法[J]. 机器人, 2014, 36(5): 627-633.
- [20] Botea A, Muller M, Schaeffer J. Near-optimal hierarchical path-finding[J]. Journal of Game Development, 2004, 1: 7-28.