

# 基于回溯的室内机器人完全遍历路径规划\*

## Full Traversal Path Planning for Indoor Robot Based on Backtracking Method

彭永昆<sup>1</sup> 徐 胜<sup>2</sup> 陈元电<sup>3</sup> 苏成悦<sup>1,2</sup> 陈 静<sup>1</sup> 罗文骏<sup>1</sup> 李艺洪<sup>1</sup> 刘 拔<sup>1</sup>

(1 广东工业大学信息工程学院, 广东 广州 510006; 2 广东工业大学物理与光电工程学院, 广东 广州 510006)

**摘要:**完全遍历路径规划算法得到越来越多的关注。提出一种基于回溯的双向完全遍历路径规划算法,根据工作环境的纵横比,选择合适的局部遍历优先级采用牛耕法完成局部覆盖,采用改进的A\*算法规划出从临界点到回溯点的无碰撞路径完成区域转移,实现工作环境的全覆盖,当回溯点个数为零时,机器人就完成了对地图的完全覆盖。仿真表明,与BA\*和B-Theta\*算法相比,在横向距离大于纵向距离如30×50的地图上,转弯次数减少15%,且能实现完全覆盖,回溯点的个数减少81%,重复路径长度减少5%。新算法有较好的参考和应用价值。

**关键词:**回溯法;牛耕法;OHA\*算法;完全遍历路径规划

**Abstract:**More and more attention has been paid to the full traversal path planning algorithm. Aiming at the problems of existing algorithms, this paper proposes a bi-directional full traversal path planning algorithm based on backtracking. According to the aspect ratio of the working environment, the algorithm selects the appropriate local traversal priority, uses the cattle ploughing method to complete the local coverage, and uses the improved A\* algorithm to plan the collision free path from the critical point to the backtracking point to complete the area transfer, so as to realize the optimization of the working environment full coverage, when the number of backtracking points is zero, the robot completes the complete coverage of the map. Simulation results show that, compared with BA\* and B-Theta\* algorithms, the number of turns is reduced by 15%, the number of backtracking points is reduced by 81%, and the length of repeat path is reduced by 5%, when the horizontal distance is greater than the vertical distance, such as 30×50. The new algorithm has good reference and application value.

**Keywords:**backtracking method, boustrophedon method, OHA\* Algorithm, full traversal path planning

随着移动机器人技术的发展,无人车得到了广泛的应用。在民用领域的清洁机器人需要具有能够完成整个室内清洁工作的能力,尽可能地在较短的时间内,以较少的行进距离完成室内的覆盖工作。算法上有生物神经激励、区域分解、模板法、随机遍历等,但这些算法存在不能实现完全覆盖、重复覆盖率高、规划速度慢、覆盖速度低等问题。全覆盖路径规划算法包括随机覆盖的算法、基于栅格地图的图搜索算法、基于图形几何的算法、基于奖励以及基于神经网络的算法等<sup>[1]</sup>。上述算法中具体有基于牛耕法和回溯法进行遍历,结合A\*算法<sup>[2-3]</sup>和Theta\*算法<sup>[4]</sup>实现从死区到回溯点的转移、基于BCD算法实现局部覆盖,结合BINN在子区域之间进行转移<sup>[5]</sup>、基于矩形区域分割地增量式全覆盖路径规划算法<sup>[6]</sup>、基于六边形分解的规划算法<sup>[7]</sup>、基于模板法并结合DWA避障算法<sup>[8]</sup>、融合生物神经激励和A\*算法完成地图全覆盖<sup>[9]</sup>。上述算法针对传统算法在覆盖率、覆盖重复率、运行效率和路径光滑程度上做了不同程度的改进,但是在面对更加复杂的工作环境时,这些算法存在运行速度慢、遍历效率低或路径不够平滑等问题。

针对全覆盖路径算法存在的问题,本文提出一种基于回溯的双向遍历全覆盖路径规划算法,根据工作环境的纵横比,该算法将选择合适的局部遍历策略,完成当前段的局部完全遍历,在机器人陷入死区时,结合回溯法,采用改进的A\*算法规划出一条从当前点到最佳回溯点的无碰撞路径,完成转移后继续遍历地图,直到回溯点用完。

### 1 传统规划方法

#### 1.1 牛耕法

牛耕单元分解法(Boustrophedon Cellular Decomposi-

tion, BCD)<sup>[10]</sup>是一种离线的区域分解的全覆盖算法,在遍历之前采用牛耕单元分解对地图进行区域分割,分割形成的不重叠单元构成了无向连通图,通过深度优先搜索(Depth First Search, DFS)对连通图进行遍历,生成单元间的遍历路径,当机器人运动到当前单元的边界时,判断当前单元遍历完毕,就可以前往路径上的下一个单元。如果当前单元已经清洁过,机器人将沿着障碍物的边缘移动到下一单元。单元内的覆盖通过牛耕法完成,即以一定顺序North、South、East、West遍历地图的四个可行方向,形成往复运动即牛耕运动。BCD算法降低了规划时计算的时间复杂度,覆盖率较高,但在障碍物较多且较零散的环境中,分解的单元数可能多且零散,单元间的遍历将更加复杂,且牛耕法运动在覆盖各个单元的过程中可能存在遗漏区域,不能完成全覆盖任务。

#### 1.2 A\*算法

A\*算法<sup>[11]</sup>是一种具备最优性、完备性和较高效率的基于栅格地图的图搜索算法,它的一个缺点就是由于要将搜索的节点储存起来,使得空间复杂度较高,时间复杂度随栅格数的增加呈指数上升。A\*基于Dijkstra算法,将贪心算法的启发式代价加入Dijkstra算法的代价函数:

$$f(s) = g(s) + h(s) \quad (1)$$

式(1)中 $g(s)$ 是从起点到当前点 $s$ 的实际距离, $h(s)$ 是从 $s$ 点到终点的启发式距离。A\*算法从起点出发,每次选择代价值 $f(s)$ 最小的栅格作为 $s$ ,随后设置为已检查点,并将 $s$ 的邻近点作为待检查点放入待检查列表,直到找到终点或待检查列表为空。通过A\*算法可以找到一条从临界点到回溯点的无碰撞路径。传统的A\*算法的启发式距离使用的是 $s$ 到终点的欧氏距

\* 广东省科技计划项目(2017A020208063);广州市科技计划项目(201804010384)

离,在实际搜索的过程中,会有大量栅格的  $f(n)$  值相同或相近,  $A^*$  算法并不能很好地处理这些相同  $f(n)$  值的栅格,最终的结果就是它们都会被搜索到,并且添加到 closed 和 open 集合中,集合中元素的增多增加了搜索的工作量,提高了  $A^*$  算法的时间复杂度。

## 2 基于回溯的双向遍历算法

### 2.1 遍历策略

针对 BCD 算法存在的问题,本文提出一种结合回溯法的双向遍历算法,基于牛耕法的基础遍历顺序,在东西向和南北向的距离不等时,选用不同的遍历策略,以减少机器人在覆盖过程中的转弯次数和重复覆盖率。算法基于栅格地图,每个栅格的边长略小于机器人的直径。在工作区域内称东西向为横向,南北向为纵向,机器人将根据地图的纵横距离长短确定覆盖方向的优先级。确定以下规则:

- 1) 对于纵向距离大于横向的区域,机器人遍历时的优先级为 North、South、East、West;
- 2) 对于横向距离大于纵向的区域,机器人遍历时的优先级为 West、East、South、North。

对于不同形状的工作区域采取不同的遍历策略,将大大减少机器人在遍历过程中的转弯次数,进而提高机器人完成区域全覆盖的效率,提高了算法的整体性能。如图 1 所示,在横向距离大于纵向距离的地图中,如在  $10 \times 20$  的地图中,NSEW 遍历优先级 a 的转弯次数是 38 次,WESN 遍历优先级 b 的转弯次数是 18 次,NSEW 的转弯次数由横向距离决定,WESN 的转弯次数由纵向距离决定,转弯次数为横向或纵向距离的  $n$  的倍数  $2(n-1)$ 。

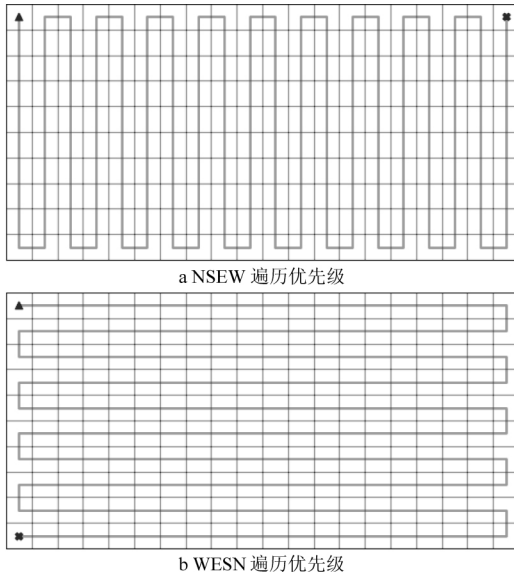


图 1 不同遍历顺序的比较

### 2.2 回溯点的选择

回溯点列表用于记录未被覆盖的栅格,以完成地图的全覆盖任务。本文提出一种回溯点的筛选方法,在上述双向遍历算法中,所有未被覆盖的栅格都将被添加到回溯点列表  $L$  中,导致回溯点列表空间占用巨大,因此需要对回溯点列表进行合适的甄选。如图 2 所示,白色为回溯点列表中未覆盖的栅格,灰色栅格可以是任意状态:

邻接点集合  $N(s) = \{S_1, S_2, \dots, S_8\}$  是当前点  $S$  在八个方向 E、NE、N、NW、W、SW、S 和 SE 上的邻接点,当  $S_3$  和  $S_7$  或  $S_1$  和  $S_5$  同时为未覆盖的栅格时, $S$  点将从回溯点列表  $L$  中剔除,最终回溯点将选择位于角落的未覆盖栅格。

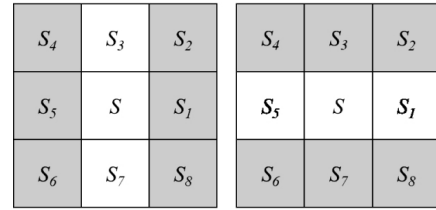


图 2 需要删除的回溯点

### 2.3 OHA \* 算法

针对  $A^*$  算法存在的搜索节点数多、路径不平滑等问题,本文提出一种基于方向启发式函数的 OHA \* 算法 (Orientation Heuristic  $A^*$ ),通过给定新的启发式函数增加关于运动方向的权重,并对规划完成的路径进行剪枝,显著地减少了  $A^*$  算法需要搜索的单元格,提高了  $A^*$  算法的运行速度,同时由于区分开了大量拥有相同代价的单元格,使得规划出来的路径更加笔直。如图 3 所示,  $S_1$ 、 $S_2$  和  $S_3$  为三个不同的栅格,  $S_{start}$  和  $S_{goal}$  分别是起点和终点,其中,它们与终点的距离关系为  $S_1 > S_2 > S_3$ ,启发式代价关系为  $h(S_1) > h(S_3) > h(S_2)$ ,  $S_2$  在位置上取得了优势。

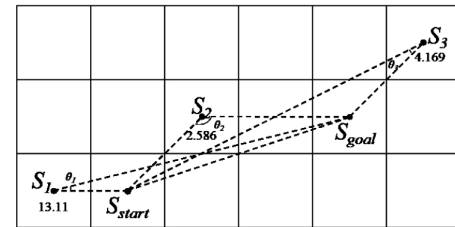


图 3 OHA \* 算法示意图

启发式函数为:

$$h'(s) = (2 + \cos\theta)h(s) \quad (2)$$

式中,  $\theta \in [0, \pi]$  值是从当前点  $S$  到起点的向量  $\vec{A} = (s, S_{start})$

和从  $S$  到终点的向量  $\vec{B} = (s, S_{goal})$  的夹角,由两个向量的内积以及模长求得:

$$\cos\theta = \begin{cases} -1, & s = S_{start} \text{ 或 } s = S_{goal} \\ \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \cdot \|\vec{B}\|}, & \text{其他} \end{cases} \quad (3)$$

$S$  与终点或起点重合时,需要令  $h'(n) = h(n)$ 。从式中可以看见,  $h'(s)$  是关于  $\theta$  值单调递减的, OHA \* 算法在选取下一个栅格时,将会更倾向于在起点和终点之间的栅格,且当栅格在起点和终点的连线的延长线上时,启发式代价的权重将达到最大,从起点到终点生成的路径将更加笔直,并使得原本拥有相同  $f(n)$  值的栅格开始区分开来,实际需要搜索的栅格大大减少,  $A^*$  算法规划的路径如图 4a 所示。

规划完成后,因为 OHA \* 算法只检查邻近的八个栅格,生成的路径不够平滑,需要对生成的路径  $P = (s_1, s_2, \dots, s_n)$  进行剪枝。剪枝算法步骤如下:

算法 1 剪枝算法

输入: 基于已知地图  $M$  规划的路径  $P = (s_1, s_2, \dots, s_n)$

输出: 剪枝后路径  $\hat{P} = (s_1, s_2, \dots, s_m)$

步骤 1.

初始化  $k=1$ , 将  $s_k$  添加到路径  $\hat{P}$ ;

步骤 2.

根据已知地图  $M$ , 找到与路径点  $s_k$  的欧氏距离最远的路径点  $s_i = (i=n, n-1, \dots, k+1)$ , 且  $s_k, s_i$  之间无障碍物;

## 步骤 3.

$s_i$  添加到路径  $\hat{P}$ ,  $k=i$ ;

## 步骤 4.

如果  $s_k$  是回溯点, 即  $s_k=s_n$ , 返回路径  $\hat{P}$ , 否则, 回到步骤 2。

当前点  $s_k$  只需要与原路径  $P$  中在  $s_k$  之后的路径点  $s_i$  ( $i=n, n-1, \dots, k+1$ ) 进行距离比较, 在这之前的路径点  $s_j$  ( $j=k-1, k-2, \dots, 0$ ) 已经在之前的循环中与  $s_k$  进行过比较, 选择距离最远的路径点, 且两点之间存在直接通路。当路径  $P$  的最后一个路径点  $s_n$  被添加到新的路径集合  $\hat{P}$  中时, 循环结束。

表 1 实验 500 次 A\*、Theta\* 和 OHA\* 算法性能比较

算法	地图尺寸	运行时间	距离代价	搜索节点数
A*	100×50	0.25	120.94	1267.55
Theta*		0.49	115.84	477.01
OHA*		0.22	115.78	659.45

剪枝之后的效果如图 4b 所示, 在可以直接到达的两点之间选择生成直接相连的路径, 生成的路径更加笔直, 路径长度进一步缩短。如图 4c 所示, 图中圆点为已搜索的栅格, 上方的▲图形为起始点, 下方的“X”图形为终点。如表 1 所示, 相比于 A\* 算法, 改进后的 OHA\* 算法生成的路径长度较传统的 A\* 算法缩短 4%, OHA\* 算法搜索的节点数较 Theta\* 算法<sup>[12-13]</sup>更多, 因而生成的路径略短, 规划时间比 Theta\* 算法缩短 55%, 比 A\* 算法缩短 12%。

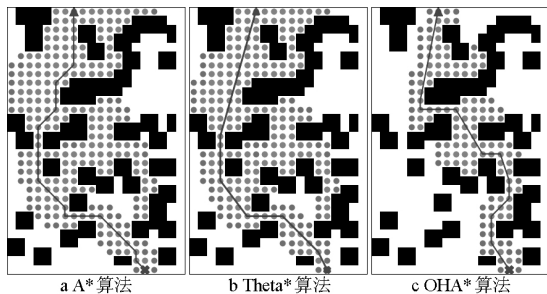


图 4 图搜索算法效果比较

当 OHA\* 被用于多个目标时, 即选择一定范围内的回溯点  $s_k \in L$  作为多目标 OHA\* 算法的目标点, 启发式函数代价为当前点与这些回溯点之间的最小距离, 可以使用欧氏距离或切比雪夫距离。随后给启发式距离加上 OHA\* 算法的权重  $(2+\cos\theta)h(s)$ , 当找到任意一个目标点时, 算法结束, 返回起点到该目标点的路径。

$$h(s) = \min_{s_k \in L} (C(s, s_k)) \quad (4)$$

## 2.4 基于回溯的双向遍历算法实现

### 2.4.1 算法原理

完整算法步骤如下:

算法 2 基于回溯的双向遍历算法

## 步骤 1.

初始化已知地图  $M$  和起点  $s_{start}$ ;

## 步骤 2.

将  $s_{start}$  设为当前点  $s$ , 从  $s$  点出发, 基于地图  $M$ , 根据地图选择合适的遍历优先级, 先将邻近的未覆盖栅格放入回溯点列表  $L$  中, 选择优先级最高的、无障碍物的且未被覆盖的邻近栅格行进一个步长, 即一个栅格, 并将该栅格从列表  $L$  中删除;

## 步骤 3.

当机器人的邻近栅格都是已覆盖或为障碍物时, 当前段覆盖运动结束, 将  $s$  点设为临界点  $s_{cp}$ , 并用前文所述的回溯点筛选方法精简列表  $L$ , 否则, 回到步骤 2;

## 步骤 4.

采用多目标 OHA\* 算法从列表  $L$  中找到回溯点  $s_{sp}$ , 作为下一段遍历的起点  $s_{start}$ , 并找到一条从  $s_{cp}$  到  $s_{sp}$  的无碰撞路径  $P$ , 机器人沿该路径移动到  $s_{cp}$ , 回到步骤 2, 如果列表为空  $L$ , 遍历结束。

通过上述分析, 完全遍历算法分为两个部分, 包括局部区域覆盖和区域转移, 局部区域覆盖将根据地图的尺寸选择合适的遍历策略进行覆盖, 区域转移采用多目标 OHA\* 算法使机器人从死区移动到最近的回溯点, 当回溯点个数为零时, 机器人就完成了对地图的完全遍历。

### 2.4.2 实验仿真

实验配置为 Intel Core i5-8500 CPU@3.00GHz, 通过与 BA\* 算法和 B-Theta\* 算法比较, 采用障碍物随机分布的不同障碍物地图, 根据两种不同的地图尺寸, 每种尺寸的地图均测试了 500 次, 以测试算法在各种可能情况下的表现, 并取平均值作为性能评估。如表 2 所示, 覆盖路径长度是覆盖运动部分的路径长度, 表示覆盖的完整程度, 相比 BA\* 算法和 B-Theta\* 算法, B-OHA\* 算法的覆盖路径长度为 1202.40 和 1204.59, 路径更长, 从图 5 可以看出, B-OHA\* 算法可以实现全覆盖。覆盖运动的分段数表示机器人需要规划回溯路径的次数, B-OHA\* 算法只有 31.89 段和 31.85 段, 段数更少。回溯路径的长度实际上是重复覆盖的部分, B-OHA\* 算法的路径长度为 201.26 和 200.93, 重复覆盖路径更短。B-OHA\* 算法在 30×50 的地图上, 转弯次数只有 172.74 次, 次数更少。在回溯点的选择上, B-OHA\* 算法的回溯点总个数只有 699.12 和 701.00 个, 个数少得多, 使得算法的运行时间只有 0.07s, 速度更快。

表 2 500 次实验仿真最终结果

	BA*		B-Theta*		B-OHA*	
	50×30	30×50	50×30	30×50	50×30	30×50
覆盖路径长度	1200.01	1201.99	1199.90	1202.00	1202.40	1204.59
分段数	38.30	39.38	38.37	39.32	31.89	31.85
转弯次数	176.07	203.04	177.40	203.85	173.67	172.74
回溯路径长度	216.04	223.03	205.81	212.36	201.26	200.93
回溯点个数	3885.12	3750.42	3865.24	3692.57	699.12	701.00
运行时间/s	0.10	0.10	0.15	0.15	0.07	0.07

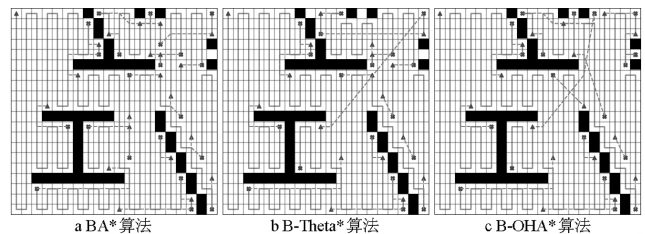


图 5 最终效果比较

## 3 结束语

经仿真表明, 相比于 BA\* 算法和 B-Theta\* 算法, B-OHA\* 算法的覆盖运动段数少 19%, 意味着机器人需要进行回溯的次数更少。在尺寸为 30×50 的地图上, 覆盖运动的转弯次数较尺寸为 50×30 的地图减少 15%, 随着地图横向距离与纵向距离的差值拉大, 这个比例也会增大, 回溯路径长度相比 B-Theta\* 算法下降 5%, 相比 B-A\* 算法下降 10%, 在不同纵横比的地图采取不同的遍历策略是有效的。回溯点选择的方法极大地减少了回溯点的个数, 相比 BA\* 算法, 回溯点个数下降 81%, 比 B-Theta\* 算法下降 81%。实验采用的是随机地图, 综合了各种复杂情况, 在复杂环境和地图尺寸较大的环境中, 由于搜索的栅格数更少, 仍能表现较好的性能。但是为了减少转弯次数, 相较于文献[14]提出的 WMF 以及文献[15]提出的优先级方法, 算法的重复覆盖率有所上升。由于规划使用的是栅格地图, 对部分形



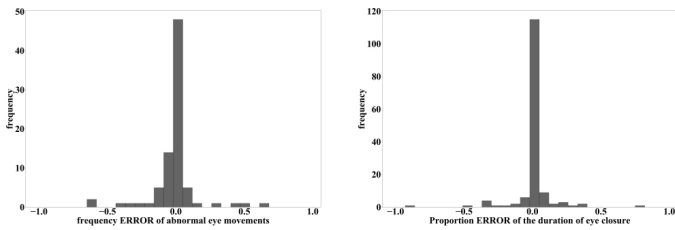
状较复杂的工作区域和障碍物,可能无法覆盖到这些障碍物的边缘。本文算法中采用的图搜索算法,在工作环境较大时,需要搜索更多的栅格,搜索的时间复杂度呈指数上升。对于存在动态障碍物的动态环境,并没有进行深入的讨论。

### 参考文献

- [1] Almadhoun R, Taha T, Seneviratne L, et al. A survey on multi-robot coverage path planning for model reconstruction and mapping [J]. Sn Applied Sciences, 2019, 1(8):
- [2] Hoang Huu V, Viet-Hung D, Laskar M N U, et al. BA\*: an online complete coverage algorithm for cleaning robots [J]. Applied Intelligence, 2013, 39(2): 217-35
- [3] Hoang Huu V, Viet-Hung D, Choi S, et al. BoB: an online coverage approach for multi-robot systems [J]. Applied Intelligence, 2015, 42(2): 157-73
- [4] Choi S, Lee S, Hoang Huu V, et al. B-Theta\*: an Efficient Online Coverage Algorithm for Autonomous Cleaning Robots [J]. Journal of Intelligent & Robotic Systems, 2017, 87(2): 265-90
- [5] 周林娜, 汪芸, 张鑫, 杨春雨. 矿区废弃地移动机器人全覆盖路径规划 [J]. 工程科学学报, 2020, 42(9): 1220-1228
- [6] Miao X, Lee J, Kang B-Y. Scalable Coverage Path Planning for Cleaning Robots Using Rectangular Map Decomposition on Large Environments [J]. IEEE Access, 2018, 6: 38200-38215
- [7] Kan X, Teng H, Karydis K. Online Exploration and Coverage Planning in Unknown Obstacle-Cluttered Environments [J]. IEEE Robotics and Automation Letters, 2020, 5(4): 5969-76
- [8] 陈超勇, 熊永根, 陶永, 等. 基于高效模板法与动态窗口法的服务机器人全覆盖路径规划方法 [J]. 高技术通讯, 2020, 30(9): 949-958
- [9] 刘晶, 姚维, 章玮. 移动机器人全覆盖路径规划算法研究 [J]. 工业控制计算机, 2019, 32(12): 52-54
- [10] Choset H. Coverage of known spaces: The boustrophedon cellular decomposition [J]. Autonomous Robots, 2000, 9 (3): 247-53
- [11] P Hart, N Nilsson, B Raphael. A formal basis for the heuristic determination of minimum cost paths [J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107
- [12] Daniel K, Nash A, Koenig S, et al. Theta\*: Any-Angle Path Planning on Grids [J]. Journal of Artificial Intelligence Research, 2010, 39(533-79)
- [13] Nash A, Koenig S, Tovey C, et al. Lazy Theta\*: Any-Angle Path Planning and Path Length Analysis in 3D [J]. Proceedings of the Twenty-Fourth Aaai Conference on Artificial Intelligence (Aaai-10), 2010, 147-54
- [14] 李楷, 陈永府, 金志勇, 等. 基于回溯法的全覆盖路径规划算法 [J]. 计算机工程与科学, 2019, 41(7): 1227-1235
- [15] 王妹婷, 齐永锋, 陆柳延, 等. 双向清洗机器人玻璃幕墙完全遍历路径规划 [J]. 机械设计与制造, 2013(11): 211-213

[收稿日期: 2021.3.4]

(上接第 32 页)



(左图特征: 眼部异常运动频率 右图特征: 长闭眼时长占比)

图 10 特征的误差分布情况

### 3 结束语

本文提出的基于几何信息的眼部异常运动分析方法,将面部运动障碍疾病患者的眼部区域运动,抽象为眼部相对开度时序波形,并采用一系列波形修正方法进行处理,从而能有效提取眼部异常运动典型特征,其特征值与人工标注的真实值相接近,说明了本文方法的有效性。这些典型特征可供眼部及面部异常运动进行后续的分析,包括疾病分型和基于 Jankovic 量表<sup>[1]</sup>的疾病严重程度评分。以几何信息及基于波形修正的方法分析眼部异常运动是本文的创新点所在,此方法可为临床上面部运动障碍疾病的计算机辅助诊断与辅助治疗提供有价值的参考。

### 参考文献

- [1] Jankovic J, Kenney C, Grafe S, et al. Relationship Between Various Clinical Outcome Assessments in Patients with Blepharospasm [J]. Movement Disorders, 2010, 24(3): 407-413
- [2] Osaki M H, Osaki T H, Garcia D M, et al. Analysis of Blink Activity and Anomalous Eyelid Movements in Patients with Hemifacial Spasm [J]. Albrecht von Graes Archiv für Ophthalmologie, 2019, 258: 669-674
- [3] Miyazaki S, Ishida A, Komatsuzaki A. A Clinically Oriented Video-based System for Quantification of Eyelid Movements. [J]. IEEE Transactions on Biomedical Engineering, 2002, 47 (8): 1088-1096
- [4] Soukupová T, Cech J. Real-time eye blink detection using facial landmarks [C] // 21st Computer Vision Winter Workshop, 2016: 1-8
- [5] Pan G, Sun L, Wu Z, et al. Eyeblink-based anti-spoofing in face recognition from a generic webcam [C] // 2007 IEEE 11th International Conference on Computer Vision. IEEE, 2007: 1-8
- [6] Zhang K, Zhang Z, Li Z, et al. Joint face detection and alignment using multitask cascaded convolutional networks [J]. IEEE Signal Processing Letters, 2016, 23(10): 1499-1503
- [7] Zadeh A, Baltrušaitis Tadas, Morency L P. Convolutional Experts Constrained Local Model for Facial Landmark Detection [C] // 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2016
- [8] Cootes T F, Taylor C J, Cooper D H, et al. Training models of shape from sets of examples [M] // BMVC92. Springer, London, 1992: 9-18
- [9] Baltrušaitis T, Zadeh A, Lim Y C, et al. Openface 2.0: Facial behavior analysis toolkit [C] // 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). IEEE, 2018: 59-66
- [10] 王婷. EMD 算法研究及其在信号去噪中的应用 [D]. 哈尔滨: 哈尔滨工程大学, 2010
- [11] Huang N E, Shen Z, Long S R, et al. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis [J]. Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences, 1998, 454(1971): 903-995

[收稿日期: 2021.1.31]