

小车远程控制及自主寻路系统的设计与实现

李晓帆 许 畅

(南京大学计算机软件新技术国家重点实验室 南京 210023) (南京大学计算机科学与技术系 南京 210023)

摘 要 智能机器人可通过传感器获取周围的地形、温度等信息,并加以分析处理而采取对应策略。然而,机器人可能由于本身不具备定位功能而只能进行简单控制,也可能因为障碍物信息采样不足而导致避障算法失败。面对这一类问题,设计并实现了一套能对智能车进行远程控制并令其自主寻路的系统。该系统通过高效的图像处理与识别对智能车自动定位;对现有避障算法进行了改进,并建立了避让策略来保证智能车总能找到一条抵达目标的安全路径。实验结果表明,该系统定位精度较高,控制与寻路功能具有一定的可靠性与实用性。

关键词 远程控制,自主寻路,安卓平台

中图分类号 TP302 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.12.022

Design and Implementation of Vehicle Remote Control and Independent Way-finding System

LI Xiao-fan XU Chang

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China)

(Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China)

Abstract Intelligent robots obtain environment information such as terrain and temperature with sensors, and then analyze and process such information to take appropriate countermeasures. However, if a robot cannot locate itself in the environment, its user can only perform simple control, and its obstacle avoidance algorithm may fail due to under-sampling of obstacle information. To address these problems, this paper proposed a system that can perform remote control and independent way-finding on smart-car robots. The system can automatically locate a smart-car robot through efficient image processing and recognition. It improves over existing algorithms and builds a collision avoidance strategy to ensure that the robot can always find a safe path to its destination. Our experimental results show that the system has high locating accuracy, and its remote control and way-finding functions are reliable and useful.

Keywords Remote control, Independent way-finding, Android platform

1 引言

智能机器人具备各式各样的信息传感器,能够实时获取环境数据;它还具有效应器——电机,能将自身行为作用于周围环境。通过智能机器人传感器与效应器,以及驱动它们的编程接口,可以实现一个基于环境感应进行自我控制的机器人自适应系统。自适应系统在机器人领域有着广阔的应用空间,已有很多工作提出了针对机器人的自适应方法^[1,2]。

然而,“智能”机器人并不如我们想象的那样智能,很多情况下,机器人并不能准确获取周围的环境信息。例如,机器人如果没有配备定位系统,就无法确定自己的方位;如果传感器数量不足,就可能因为获取的环境信息不全而对附近障碍的分布情况产生误判,从而导致碰撞事故的发生。当多个机器人同时运行时,情况会更加复杂。例如,在 t_1 时刻,机器人 R_1 检测到 A 点的机器人 R_2 ;在 t_2 时刻 R_2 行驶到 B 点,如果 R_1 依然保留着 A 点的障碍物信息,一个可行的方向可能会因此被错误地否决。

机器人的上述缺陷导致对其进行自适应系统开发比较困难。针对这一问题,本文设计并实现了一个能对智能车进行远程控制并自主寻路的自适应系统。主要贡献分为以下 3 点:

- (1)通过 Android 设备对小车进行远程控制。
- (2)能对小车准确定位,并在 Android 屏幕上实时显示位置与轨迹。
- (3)通过点击屏幕设定目标,小车就能自主寻路,支持两车同时运行。

本文第 2 节介绍远程控制及其依赖功能的实现过程;第 3 节介绍寻路算法与避让机制;第 4 节对两个功能进行了实验,并对结果进行分析评估;第 5 节介绍相关工作;最后对整个工作进行总结,并说明未来工作。

2 远程控制

系统通过 Android 设备向智能车发送指令,进行远程控制;同时利用天花板上摄像头不断采集的环境信息,绘制出地

到稿日期:2015-02-12 返修日期:2015-03-28 本文受国家重点基础研究发展计划(973)(2015CB352202),国家自然科学基金(61472174, 91318301, 61321491, 61361120097)资助。

李晓帆(1992-),男,硕士,主要研究领域为普适计算中的上下文管理, E-mail: leslie. ido@gmail. com; 许 畅(1977-),男,博士,副教授, CCF 会员,主要研究领域为软件工程、软件测试与分析、普适计算上下文管理、物联网、射频识别等。

图,再分析地图从中识别小车,对其进行准确定位。

2.1 指令缓冲区

Android 设备与智能车之间通过建立 TCP 连接进行通信。用户指令被封封装成数据包发送到小车。由于小车接收到新指令时会中断当前指令,因此需在 Android 设备与小车两端分别建立指令缓冲区。

Android 程序为每辆小车维护一个发送缓冲区和一个接收缓冲区,每个缓冲区各有一个发送与接收线程进行管理。用户发送命令时,程序将对对应指令放入发送缓冲区。当缓冲区不为空时,发送线程按照 FIFO 原则将指令发送给小车。接收线程收到小车指令后放入接收缓冲区,一个 handler 线程从该缓冲区取出指令并作相应处理。

ARM 程序端(即小车端)的指令缓冲区与 Android 程序端相同。handler 线程处理指令时,会根据其参数预估执行时间 X 。handler 将命令交给底层控制器后,会等待 X 时间再从接收缓冲区取出下一条指令。通过建立等待机制,小车能够顺序执行指令序列,且不中断当前指令。

2.2 地图绘制

为了在 Android 屏幕上显示小车位置,需要利用天花板上的 9 个摄像头对其进行定位。令摄像头每秒抓拍一张图像(等于每秒 1 帧的视频),在满足定位精度与实时性的同时,避免了处理实时视频流的巨大开销。

由于抓拍的原始图像存在畸变,因此首先需要标定摄像头。采用了棋盘标定算法^[3],根据棋盘格点在空间平面中三维点到摄像头平面中像素坐标的映射关系,求解出摄像头的畸变矩阵,再对原始图像应用畸变矩阵完成矫正。对 9 个摄像头抓拍的图像依次进行上述处理,并根据实际物理位置拼接得到完整地图,如图 1 所示。

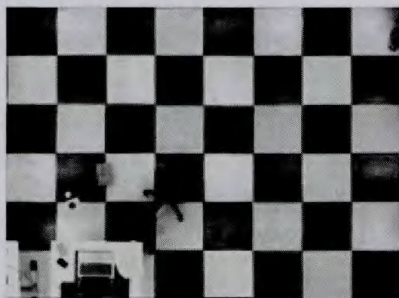


图 1 完整地图

2.3 小车定位

获得地图后,通过分析地图对小车定位。图像识别一般需要先提取图像特征。由于可以自行设计特征鲜明的识别图案置于车顶,因此不必特意提取特征并大量训练数据集,而是直接用图案的所有特征制作模板,与源图像进行对比匹配。

模板匹配是通过将模板图像 T 在源图像 S 上滑动,并按照给定的匹配度准则比较 T 与 S 重合部分的像素值,最终得到匹配度最大的坐标点。这种方法的优点是简单直接,缺点是不具有伸缩不变性与旋转不变性。当源图像经过缩放或旋转后,算法准确度会降低。在实验环境中,由于小车与摄像头距离较远,车顶的匹配图案在地图中的尺寸变化可以忽略,因此只需考虑图案的旋转问题。根据圆的旋转不变性,设计了两个匹配图案,分别用于识别小车 1(图 2 左)和小车 2(图 2 中)。两幅图案的上半部为车头模板,下半部为车尾模板。由图 2 右可以看出,当小车转向引起匹配图案在地图中发生旋

转时,模板依然可以和图案很好地匹配。



图 2 匹配图案与结果

2.4 结合定位的地图优化

在实际运行中,如果每次都对所有图像进行处理,会耗费大量时间。利用上次对小车的定位,可以划定这次小车位置的大致范围,从而只需处理该范围的摄像头抓拍的图像。

优化算法将整张地图划分为 25 个区域(见图 3),如实线所示。虚线表示 9 张图像的拼接边界,25 个区域中的数字表示当小车出现在该区域时,需要处理分析的图像数目。例如,若小车定位到标识为 4 的区域(简称 4 型区),则需分析该区域左上、右上、左下、右下 4 张图像。

1	2	1	2	1
2	4	2	4	2
1	2	1	2	1
2	4	2	4	2
1	2	1	2	1

图 3 区域划分

优化算法进一步将每张图像分为 4 块,分情况截取分析。若小车位于 1 型区,则分析整张图像;若位于 2 型区,则截取与该区域毗邻的图像的一半进行分析。例如,小车位于第一个 2 型区,则截取第一张图像的右半与第二张图像的左半。对 4 型区的处理方式类似。

经过优化后,无论小车位于哪个区域,需要分析的图像范围都等于一张图像,大大减少了图像处理的时间开销。

3 自主寻路

3.1 避障算法

本文的避障算法采用 VFH⁺^[8]的核心思想。算法使用二维的笛卡尔直方图栅格表示障碍物环境,每个栅格单元存储一个确信值(Certainty Value, CV)来表示算法对该位置存在障碍物的确信度。当机器人在移动时,以机器人中心点(Robot Center Point, RCP)为中心建立活动窗口(见图 4)。活动窗口中的栅格单元称为活动单元,每个活动单元 $C_{i,j}$ 都根据其 CV 值与到 RCP 的距离计算出对应障碍值。算法 1 显示了逐步选出方向的过程。

算法 1 避障算法

输入:直方图栅格,小车方位,目标位置

输出:前进方向

1. 建立活动窗口 w_s
2. 平分 w_s 为 n 个扇区
3. For $k=1$ to n
4. 对扇区 k 所有活动单元的障碍值求和 s_k
5. 若 s_k 小于设定阈值,则标记扇区 k
6. 筛选出所有标记的区间(openings)
7. 初始化候选方向集合 C
8. For each opening O_i

9. If $o_i \cdot \text{range} < T_{\text{opening}}$
10. 将 o_i 的中间扇区加入 C
11. Else
12. 将距离 o_i 左右边界 $T_{\text{opening}}/2$ 个扇区的两个扇区加入 C
13. If 目标方向 $\in o_i$
14. 将目标扇区加入 C
15. 利用代价函数在 C 中选出代价最小方向 k_s

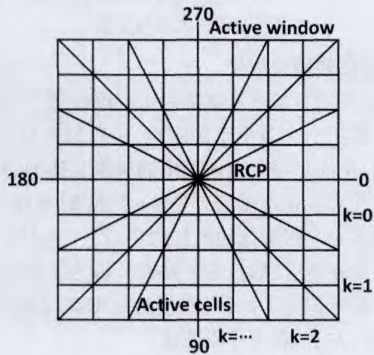


图4 活动窗口

本文中所使用的智能车因为传感器数量较少,导致采样得到的障碍物信息中噪点较多,无法直接使用 VFH+。下面针对这些不足进行改进。

利用代价函数选出方向 k_s 后,与小车当前方向进行比较。若 $k_s \neq k_c$,则控制小车原地转向至 k_s ,否则小车前进指定距离。在选出方向后,小车不直接转向并向前行驶的原因有两点:1)小车转向存在误差,若转向后直接前进,可能会导致小车向错误的方向行驶;2)小车传感器无法同时探测所有方向,若选择的方向恰好处于探测死角,那么转向后直接前进可能导致小车与无法探测到的障碍物相撞。

由于实验环境为静态环境,机器人所处位置不可能存在障碍物,将 RCP 周围单元的 CV 值清零,并置标志位。为提高算法效率,设定了 CV 值的上界 CV_{max} ,并对整个栅格进行定期更新:

$$c_{i,j}^n = \begin{cases} c_{i,j}^{n-1} - 1, & 0 < c_{i,j}^{n-1} < CV_{\text{max}} \\ c_{i,j}^{n-1}, & \text{otherwise} \end{cases}$$

其中, $c_{i,j}$ 表示未置标志位的栅格 $C_{i,j}$ 单元的 CV 值。更新规则表明:当一个单元的 CV 值不为 0 或 CV_{max} 时,表示该位置存在障碍物的确信度并不高,可能是噪点,因此 CV 值会不断衰减。随着时间的推移与小车活动范围的扩大,栅格中明确的单元(指置标志位或 CV 值为 0 或 CV_{max})会逐渐增多,噪点的影响也会越来越小,算法的执行效果将不断提升。

3.2 避让机制

避障算法是为静态环境中小车的自适应逻辑所设计的。如果两辆小车同时寻路,一辆车上的算法可能由于另一辆车的移动而失效,因此设计了一套避让机制,令小车能够对另一辆车做出正确反应。

Android 设备不断接收两车最新的位置信息,并转发给小车。每辆小车获得其他小车的信息后,在栅格中将另一小车,附近的单元清零并置标志位。这样传感器即使检测到另一小车,也不会将障碍物信息存入栅格。两车之间存在一个安全距离 d_{safe} 。当车距大于 d_{safe} 时,避障算法正常执行。由于小车附近单元的 CV 值恒为 0,实际上过滤了另一辆小车的信息。当车距小于 d_{safe} 时,如果小车在读取另一小车附近单

元的 CV 值(实际为 0),则避障机制会将这些值置为 CV_{max} ,让避障算法误以为该车位置上一定存在障碍物,因此必然会避开它。

采取保守的避让方式来避免两车选择相向的避让方向:当车距小于 d_{safe} 时,令一辆车停下,另一辆进行正常避让;当车距拉开至 d_{safe} ,激活停止的小车。小车在行驶与停止的状态间切换时,将状态通知 Android 设备。Android 设备若检测到两车均停止且并非都抵达目标,则激活未达目标的小车。

4 实验与分析

4.1 实验环境

实验场景用箱子围成墙壁,利用履带车作为单个障碍物。两辆智能车尺寸为 $32\text{cm} \times 32\text{cm}$,前、后、左、右 4 面各有一个超声波发射器和两个超声波接收器。小车运行速度设定为 30cm/s 。

4.2 远程控制实验

远程控制实验测试小车能否遵循 Android 程序所下达的指令,做出相应动作。在 Android 程序的控制界面(见图 5 左)输入以下指令序列:

Forward $\times 5 \rightarrow$ Right $\times 2 \rightarrow$ Forward $\times 6 \rightarrow$ Right $\times 3 \rightarrow$ Backward $\times 7$

小车行驶结束后在 Android 程序的地图界面显示的方位与轨迹如图 5 右部所示。地图中的圆表示小车位置,竖杆表示小车方向,灰色线条为小车的行驶轨迹。地图上方显示了小车 id、像素坐标以及方向。



图5 Android应用界面

4.3 自主寻路实验

实验给两辆小车指定目标,同时进行寻路。实验场景俯视图如图 6 左部所示,两小车在 Android 地图上的对应位置如图 6 右部所示。通过点击地图下方的按钮,依次切换至小车 1 与小车 2,再点击地图上另一辆车的位置为目标,两小车即进入寻路阶段。

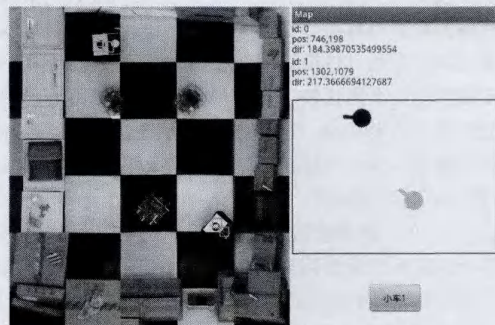


图6 初始位置

4.4 结果分析

对于远程控制实验,在 Android 程序的控制界面快速输入指令序列后,由于指令缓冲区的存在,小车会逐条取出指令并执行。由图 5 右部中小车的行驶轨迹可以看出,小车按正确的顺序执行了 Android 程序所下达的指令序列。

自主寻路实验测试了两辆小车同时寻路的避障与避让功能。图 7 显示两小车经过寻路,都抵达了对方的初始位置。从图 7 右图中发现,两车在目标附近的行驶轨迹都比较杂乱,这是因为利用模板匹配对小车进行定位时会有误差,且小车执行指令时也会存在偏差,两者的积累导致小车在到达目标点时需要不断矫正方位,反复尝试靠近目标。

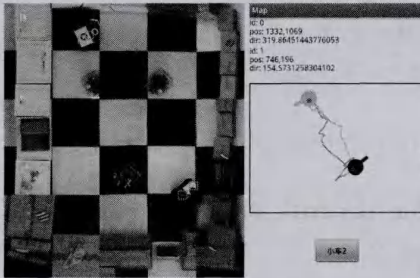


图 7 最终位置

两辆小车的栅格如图 8 所示。由小车 1 栅格(见图 8 左部)看出,CV 值集中在小车初始位置与目标位置。这是因为小车 1 的初始位置附近障碍物比较密集,除两面有墙外,一侧还有两个障碍物,因此小车 1 原地转向多次才找到两个障碍物之间的缺口。而到达目标附近后,由于前面提到的定位与指令误差因素,小车 1 在目标附近折返多次,因而对此处的障碍物信息采样较多。小车 2(见图 8 右部)的情况则不相同,CV 值仅集中在目标位置。寻路开始后,小车 2 迅速找到了前方的宽缺口,驶离了初始地点,因而它对左侧的障碍物几乎没有采样信息。到达目标点附近,同样基于误差因素,小车 2 对目标附近的墙壁与障碍物反复扫描多次,导致 CV 值很高。

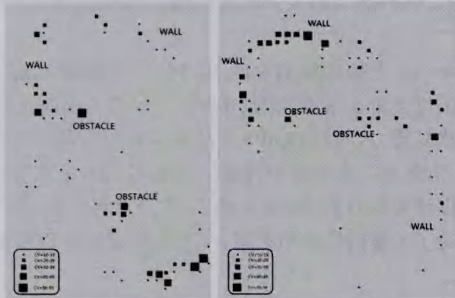


图 8 栅格示意图

5 相关工作

Silva 等人^[4]提出了感知需求的概念,用以感知自适应系统中其他需求满足与否;Ramirez 等人^[5]提出面向自适应系统的设计模式,便于功能与自适应逻辑分离;Georgas 等人^[6]在不同机器人案例中对应用基于策略的自适应框架进行了可行性研究。

在机器人的避障算法方面,Borenstein 等人^[9]用栅格表示环境中的障碍物信息,利用势场法选择机器人的行进方向,

但其缺陷是无法通过狭窄通道,在障碍物前震荡。因此 Borenstein 等人又提出 VFH^[7],即将栅格等分为多个扇区,通过多步数据简化,选出合适扇区作为前进方向。后来 Ulrich 等人提出 VFH+ 算法^[8],它考虑到了机器人自身的体积与转弯半径,避免了选择实际不可行的行驶方向。

本文中的自主寻路算法以 VFH+ 为核心,针对智能车传感器数量不足、定位耗时长长的实际情况,设计了衰减函数与先转再走策略,减少了栅格中的噪点,避免小车因信息不足而选择错误的行进方向。

结束语 本文设计了一个远程控制小车并令其自主寻路的系统。它将 Android 移动设备与嵌入式系统相结合,实现了通过 Android 设备灵活地对小车进行远程控制。系统在小车上实现了以 VFH+ 为核心的避障算法,并建立了多车运行的避让机制。用户可以方便地点击 Android 设备的屏幕来为不同小车指定目标,并从屏幕上实时观测小车的方位与行驶轨迹。

目前该系统还存在一些不足有待加强,如 Android 设备端的远程控制功能比较简单,通过抓拍、处理图像来定位小车耗时较多。在将来的工作中会加入更多控制指令,丰富功能,并给小车配置定位工具,尝试其他定位方法来缩短定位用时。

参考文献

- [1] Vergaro E, Casadio M, Squeri V, et al. Self-adaptive robot training of stroke survivors for continuous tracking movements[J]. *Journal of Neuroengineering and Rehabilitation*, 2010, 7(13): 1-12
- [2] Wang J S, Lee C S G. Self-adaptive recurrent neuro-fuzzy control of an autonomous underwater vehicle[J]. *IEEE Transactions on Robotics and Automation*, 2003, 19(2): 283-295
- [3] Zhang Z. A flexible new technique for camera calibration[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, 22(11): 1330-1334
- [4] Silva Souza V E, Lapouchnian A, Robinson W N, et al. Awareness requirements for adaptive systems[C]//*Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-managing Systems*. ACM, 2011: 60-69
- [5] Ramirez A J, Cheng B H C. Design patterns for developing dynamically adaptive systems[C]//*Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems*. ACM, 2010: 49-58
- [6] Georgas J C, Taylor R N. Policy-based self-adaptive architectures: a feasibility study in the robotics domain[C]//*Proceedings of the 2008 International Workshop on Software Engineering for Adaptive and Self-managing Systems*. ACM, 2008: 105-112
- [7] Borenstein J, Koren Y. The vector field histogram-fast obstacle avoidance for mobile robots[J]. *IEEE Transactions on Robotics and Automation*, 1991, 7(3): 278-288
- [8] Ulrich I, Borenstein J. VFH+: Reliable obstacle avoidance for fast mobile robots[C]//*Proceedings 1998 IEEE International Conference on Robotics and Automation*, 1998. IEEE, 1998, 2: 1572-1577
- [9] Borenstein J, Koren Y. Real-time obstacle avoidance for fast mobile robots[J]. *IEEE Transactions on Systems, Man and Cybernetics*, 1989, 19(5): 1179-1187