Roland Siegwart · Illah R. Nourbakhsh

自主移动机器人导论

〔美〕 R·西格沃特 I·R·诺巴克什 著

李人厚 译

Introduction to Autonomous Mobile Robots

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

Introduction to Autonomous Mobile Robots

自主移动机器人导论

〔美〕 R·西格沃特
I·R·诺巴克什 著

Roland Siegwart
Swiss Federal Institute of Technology, Lausanne

Illah R. Nourbakhsh
Carnegie Mellon University

李人厚 译

西安交通大学出版社
Xi'an Jiaotong University Press

内 容 提 要

任何成功的机器人的设计涉及到运动学、信号分析、信息论、人工智能和概率论等多个不同学科的综合。为此，本书提出了在一系列交互模块中，使移动性成为可能的技术和制作工艺。随着各章内容的深入，覆盖了移动机器人学的各个方面，包括硬件设计、轮子设计、运动学分析、传感器、感知、定位、作图，以及机器人控制体系结构。本书把移动机器人的所有方面集合成一个整体，既适合作为我国各高等院校自动化、机器人学等专业的研究生教材，也可提供其他相关专业的科技人员参考。

Roland Siegwart and Illah R. Nourbakhsh
Introduction to Autonomous Mobile Robots
ISBN:0-262-19502-X
Copyright©2004 by Massachusetts Institute of Technology
Original language published by Massachusetts Institute of Technology. All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.
Simplified Chinese translation edition jointly published by Massachusetts Institute of Technology and Xi'an Jiaotong University Press.

本书中文简体字翻译版由美国 MIT 出版社授权西安交通大学出版社出版发行。未经出版者预先书面许可，不得以任何方式复制和抄袭本书的任何部分。
陕西省版权局著作权合同登记号：25－2005－069 号

图书在版编目(CIP)数据

自主移动机器人导论 /（美）西格沃特（Siegwart，R.），（美）诺巴克什（Nourbakhsh，I. R.）著；李人厚译. —西安：西安交通大学出版社，2006.8
书名原文：Introduction to Autonomous Mobile Robots
ISBN 7-5605-2161-4

Ⅰ. 自…　Ⅱ. ①西…②诺…③李…　Ⅲ. 移动式机器人-简介　Ⅳ. TP242

中国版本图书馆 CIP 数据核字(2006)第 097231 号

书　　名	自主移动机器人导论
著　　者	〔美〕R·西格沃特　I·R·诺巴克什
译　　者	李人厚
出版发行	西安交通大学出版社
地　　址	西安市兴庆南路 25 号(邮编：710049)
电　　话	(029)82668357　82667874(发行部) (029)82668315　82669096(总编办)
电子邮件	xjtupress@163.com
印　　刷	陕西宝石兰印务有限责任公司
字　　数	266 千字
开　　本	727 mm ×960 mm　1/16
印　　张	21
印　　次	2006 年 9 月第 1 版　2006 年 9 月第 1 次印刷
印　　数	0 001～3 000
书　　号	ISBN 7-5605-2161-4 /TP·432
定　　价	37.00 元

版权所有　侵权必究

序　言

移动机器人学是一个年轻的领域。它的基础包括了许多工程和科学学科，从机械、电气和电子工程到计算机、认知和社会科学。这些主要领域，各有它们介绍性的教科书部分，用以激励和鼓舞新的学生，为他们以后的高级课程和研究做准备。著述本书的目的是给移动机器人学提供预备性的指导。

本书对移动机器人学的基础进行了介绍，其范围包括构成本领域的机械、电机、传感器、感知和认知等各个方面。我们知道，专题学术讨论会的论文集和期刊发表的论文，可以向新学生提供移动机器人学所有方面的技术发展水平的简单印象。但在这里，我们的目的是提供基础知识——该领域的正式介绍。正当技术发展的前沿因所有移动机器人学的分支学科迅速进步而向前推进的时候，书中的形式描述和分析，会证明是有用的。

我们希望，本书能向机器人学本科和研究生，提供他们整个职业生涯中，评估甚至评审移动机器人的方案和产品所需要的基础知识和分析工具。总体而言，本教科书作为概论性的移动机器人学的课本，既适合于本科生，又适合于研究生。个别章节，如感知或运动学，可以作为有关机器人学的特殊子领域中更专门的课程的综述。

本书的起源横跨大西洋。作者在美国斯坦福大学、瑞士联邦理工大学（苏黎世）、美国卡内基梅隆大学和瑞士联合工业大学（洛桑）为本科生和研究生教授过移动机器人学课程。它们的课程资料和讲义的组合形成了本书最初的版本。我们已综合了各自的笔记，提出了总的架构，并且在此教材决定出版之前又试教了两年。

本书组织的纵览和各章概要请参阅 1.2 节。

最后，对教师和学生而言，我们希望，对机器人学众多的发展，本书证实是一个具有富有成果的起始点，那就是对我们莫大的奖励了。

目　　录

序言
第 1 章　引言
1.1　引言 …… (1)
1.2　本书综述 …… (11)

第 2 章　运动
2.1　引言 …… (13)
2.1.1　运动的关键问题 …… (17)
2.2　腿式移动机器人 …… (18)
2.2.1　腿的构造与稳定性 …… (18)
2.2.2　腿式机器人运动的例子 …… (21)
2.3　轮式移动机器人 …… (31)
2.3.1　轮子运动:设计空间 …… (31)
2.3.2　轮子运动:实例研究 …… (39)

第 3 章　移动机器人运动学
3.1　引言 …… (47)
3.2　运动学模型和约束 …… (48)
3.2.1　表示机器人的位置 …… (49)
3.2.2　前向运动学模型 …… (51)
3.2.3　轮子运动学约束 …… (54)
3.2.4　机器人运动学约束 …… (61)
3.2.5　举例:机器人运动学模型和约束 …… (63)
3.3　移动机器人的机动性 …… (67)
3.3.1　活动性的程度 …… (67)

3.3.2 可操纵度……(72)
3.3.3 机器人的机动性……(73)
3.4 移动机器人工作空间……(75)
3.4.1 自由度……(75)
3.4.2 完整机器人……(76)
3.4.3 路径和轨迹的考虑……(78)
3.5 基本运动学之外……(82)
3.6 运动控制……(82)
3.6.1 开环控制……(83)
3.6.2 反馈控制……(84)

第4章 感知

4.1 移动机器人的传感器……(91)
4.1.1 传感器分类……(91)
4.1.2 表征传感器的特性指标……(93)
4.1.3 轮子/电机传感器……(100)
4.1.4 导向传感器……(101)
4.1.5 基于地面的信标……(104)
4.1.6 有源测距……(107)
4.1.7 运动/速度传感器……(119)
4.1.8 基于视觉的传感器……(120)
4.2 表示不确定性……(149)
4.2.1 统计的表示……(149)
4.2.2 误差传播:对不确定的测量进行组合……(153)
4.3 特征提取……(155)
4.3.1 基于距离数据的特征提取(激光、超声和基于视觉测距)……(158)
4.3.2 基于可视表象的特征提取……(166)

第5章 移动机器人的定位

5.1 引言……(185)
5.2 定位的挑战:噪声和混叠……(186)
5.2.1 传感器噪声……(187)
5.2.2 传感器混叠……(188)
5.2.3 执行器噪声……(189)

5.2.4 里程表位置估计的误差模型 …… (191)
5.3 定位或不定位:基于定位的导航与编程求解的对比 …… (196)
5.4 信任度的表示 …… (199)
5.4.1 单假设信任度 …… (199)
5.4.2 多假设信任度 …… (202)
5.5 地图表示方法 …… (205)
5.5.1 连续的表示方法 …… (205)
5.5.2 分解策略 …… (208)
5.5.3 发展水平:地图表示方法的最新挑战 …… (215)
5.6 基于概率地图的定位 …… (218)
5.6.1 引言 …… (218)
5.6.2 马尔可夫定位 …… (221)
5.6.3 卡尔曼滤波器定位 …… (235)
5.7 定位系统的其他例子 …… (251)
5.7.1 基于路标的导航 …… (252)
5.7.2 全局唯一定位 …… (253)
5.7.3 定位信标系统 …… (255)
5.7.4 基于路由的定位 …… (257)
5.8 自主地图的构建 …… (257)
5.8.1 随机构图的技术 …… (258)
5.8.2 其他的构图技术 …… (261)

第6章 规划与导航

6.1 引言 …… (265)
6.2 导航能力:规划和反应 …… (266)
6.2.1 路径规划 …… (268)
6.2.2 避障 …… (281)
6.3 导航的体系结构 …… (300)
6.3.1 代码重用与共享的模块性 …… (300)
6.3.2 控制定位 …… (301)
6.3.3 分解技术 …… (302)
6.3.4 实例研究:分层机器人结构 …… (308)

参考文献

移动机器人的范围覆盖着从火星探索器上的遥控旅居者到巴黎超市的清洁机器人。本书向学生和其他有兴趣的读者提供移动性技术的概述。移动性就是容许移动机器人穿越现实环境而进行运动并完成其任务——包括运动、传感、定位和运动规划的一种机制。本书讨论了移动机器人学的各个方面，包括硬件设计、轮子设计、运动学分析、传感器和感知、定位、作图以及机器人控制体系结构。

任何成功的机器人的设计涉及到多个不同学科的综合，这些学科有：运动学、信号分析、信息论、人工智能和概率论。为此，本书提出了在一系列交互模块中，使移动性成为可能的技术和制作工艺。随着本书细节从低级到高级的发展，各章覆盖了移动性的不同方面。前两章揭示了低级的运动能力，检验机器人的轮子和腿以及运动学的原理。接着，深入观察感知，包括许多“现货供应”的传感器和传感数据互操作的分析。最后两章考虑了定位和认知的较高水平的挑战性问题，讨论了成功的定位策略、自动作图和导航能力。本书把移动机器人的所有方面集合成一个整体，可以用作课程的教科书，或者为该领域的初学者提供使用的工具。

作者简介：

R·西格沃特，瑞士洛桑联邦理工大学教授，自主系统实验室主任。

I·R·诺巴克什，卡内基梅隆大学计算机学院机器人学研究所副教授。

“该书易于阅读，组织合理。它将机器人的功能结构作为全书的纲要，然后在章节里阐述各个组成部分，这种呈述的方法具有突出的优点。我认为，作者已经实现了他们的目标，不论是初学者还是高年级学生，都会从书中获得如何能给机器人赋予移动性的清晰概念。”

——R·查梯拉，法国科学研究中心系统结构和分析实验室主任

(Raji Chatila, LAAS-CNRS, France)

策划编辑／赵丽平　责任编辑／鲍　媛
封面设计／阎　亮　版式设计／程文卫

ISBN 7-5605-2161-4

9 787560 521619 >

ISBN 7-5605-2161-4/TP·432

定价：37.00元

[General Information]
书名=自主移动机器人导论
页数=327
SS号=11736098

封面
书名
版权
前言
目录
第1章　引言
1.1　引言
1.2　本书综述
第2章　运动
2.1　引言
2.1.1　运动的关键问题
2.2　腿式移动机器人
2.2.1　腿的构造与稳定性
2.2.2　腿式机器人运动的例子
2.3　轮式移动机器人
2.3.1　轮子运动：设计空间
2.3.2　轮子运动：实例研究
第3章　移动机器人运动学
3.1　引言
3.2　运动学模型和约束
3.2.1　表示机器人的位置
3.2.2　前向运动学模型
3.2.3　轮子运动学约束
3.2.4　机器人运动学约束
3.2.5　举例：机器人运动学模型和约束
3.3　移动机器人的机动性
3.3.1　活动性的程度
3.3.2　可操纵度
3.3.3　机器人的机动性
3.4　移动机器人工作空间
3.4.1　自由度
3.4.2　完整机器人
3.4.3　路径和轨迹的考虑
3.5　基本运动学之外
3.6　运动控制
3.6.1　开环控制
3.6.2　反馈控制
第4章　感知
4.1　移动机器人的传感器
4.1.1　传感器分类
4.1.2　表征传感器的特性指标
4.1.3　轮子／电机传感器
4.1.4　导向传感器
4.1.5　基于地面的信标
4.1.6　有源测距
4.1.7　运动／速度传感器
4.1.8　基于视觉的传感器
4.2　表示不确定性
4.2.1　统计的表示
4.2.2　误差传播：对不确定的测量进行组合
4.3　特征提取
4.3.1　基于距离数据的特征提取（激光、超声和基于视觉测距）
4.3.2　基于可视表象的特征提取
第5章　移动机器人的定位
5.1　引言
5.2　定位的挑战：噪声和混叠
5.2.1　传感器噪声
5.2.2　传感器混叠
5.2.3　执行器噪声
5.2.4　里程表位置估计的误差模型
5.3　定位或不定位：基于定位的导航与编程求解的对比

5.4　信任度的表示
5.4.1　单假设信任度
5.4.2　多假设信任度
5.5　地图表示方法
5.5.1　连续的表示方法
5.5.2　分解策略
5.5.3　发展水平：地图表示方法的最新挑战
5.6　基于概率地图的定位
5.6.1　引言
5.6.2　马尔可夫定位
5.6.3　卡尔曼滤波器定位
5.7　定位系统的其他例子
5.7.1　基于路标的导航
5.7.2　全局唯一定位
5.7.3　定位信标系统
5.7.4　基于路由的定位
5.8　自主地图的构建
5.8.1　随机构图的技术
5.8.2　其他的构图技术
第6章　规划与导航
6.1　引言
6.2　导航能力：规划和反应
6.2.1　路径规划
6.2.2　避障
6.3　导航的体系结构
6.3.1　代码重用与共享的模块性
6.3.2　控制定位
6.3.3　分解技术
6.3.4　实例研究：分层机器人结构

第1章　引　言

1.1　引言

至今，在工业制造中，机器人学已取得了最伟大的成功。机器人手臂或**机械手**，构成了20亿美元的工业产值。在装配线中，把机器人的肩膀用螺栓固定在特定位置上，机器人的手臂可以极快速度和极高精度移动，完成重复性任务，如点焊和喷漆等(图1.1)。在电子工业中，机械手以超人的精度安放装在表面的元件，可制造便携式电话和笔记本电脑。

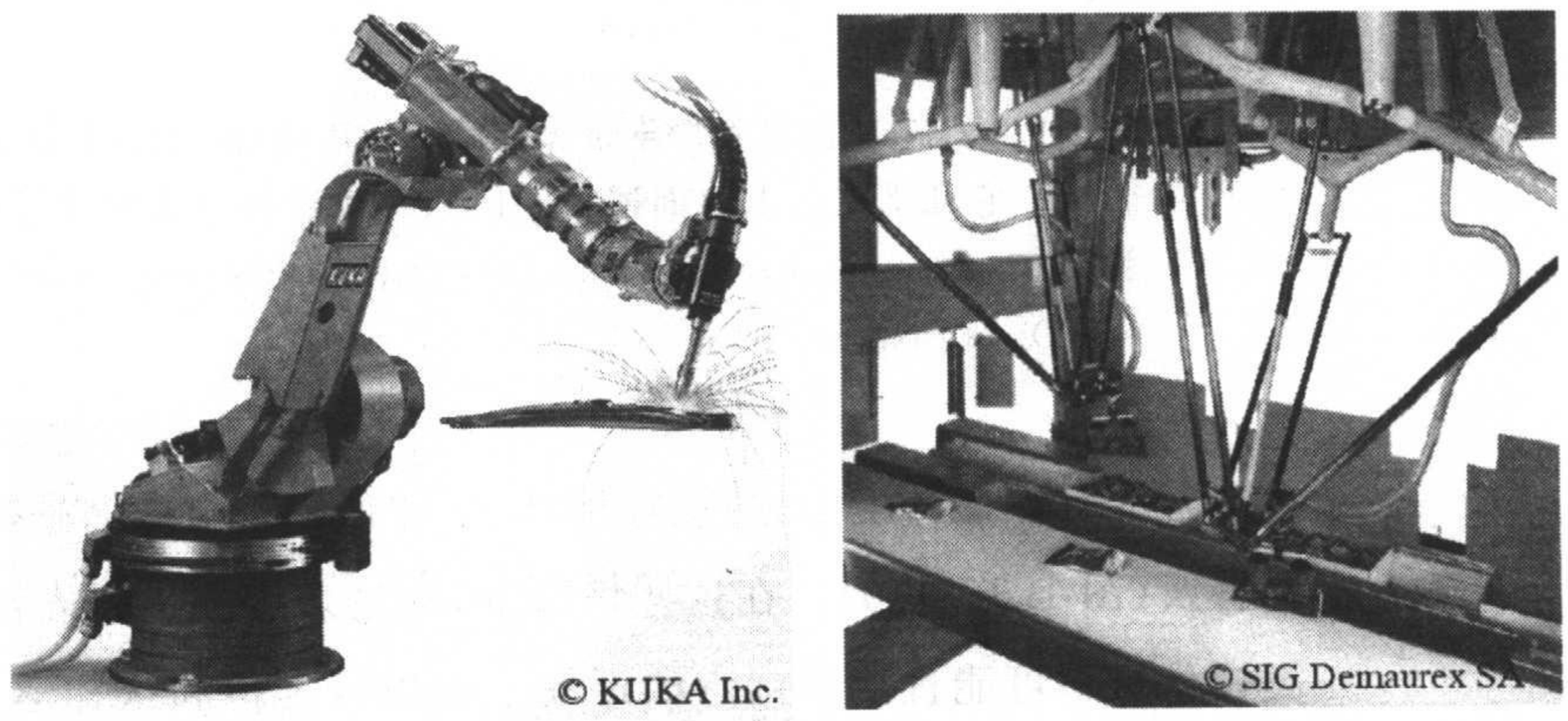


图1.1　在巧克力包装时自动装配厂焊接机器人KUKA和SIG Demaurex SA(EPFL[140]创作)的平行机器人Delta的照片

但是，对于所有的这些成功应用，商用机器人存在着一个根本的缺点：缺乏机动性。固定的机械手被插销在固定地方，其运动范围是有限的。相反，移动机器人能够行走，穿过整个制造工厂，灵活地在它最有效的地方施展它的才能。

本书着重于研究机动性技术：移动机器人如何能够无监督地移动，穿过现实世界环境完成它的任务？第一个挑战的问题是运动本身，机器人应该怎样运动？什么是它的特殊运动机构，使它优于别的运动机构？

恶劣环境，诸如在金星，引发了更为不寻常的运动机构(图 1.2)。

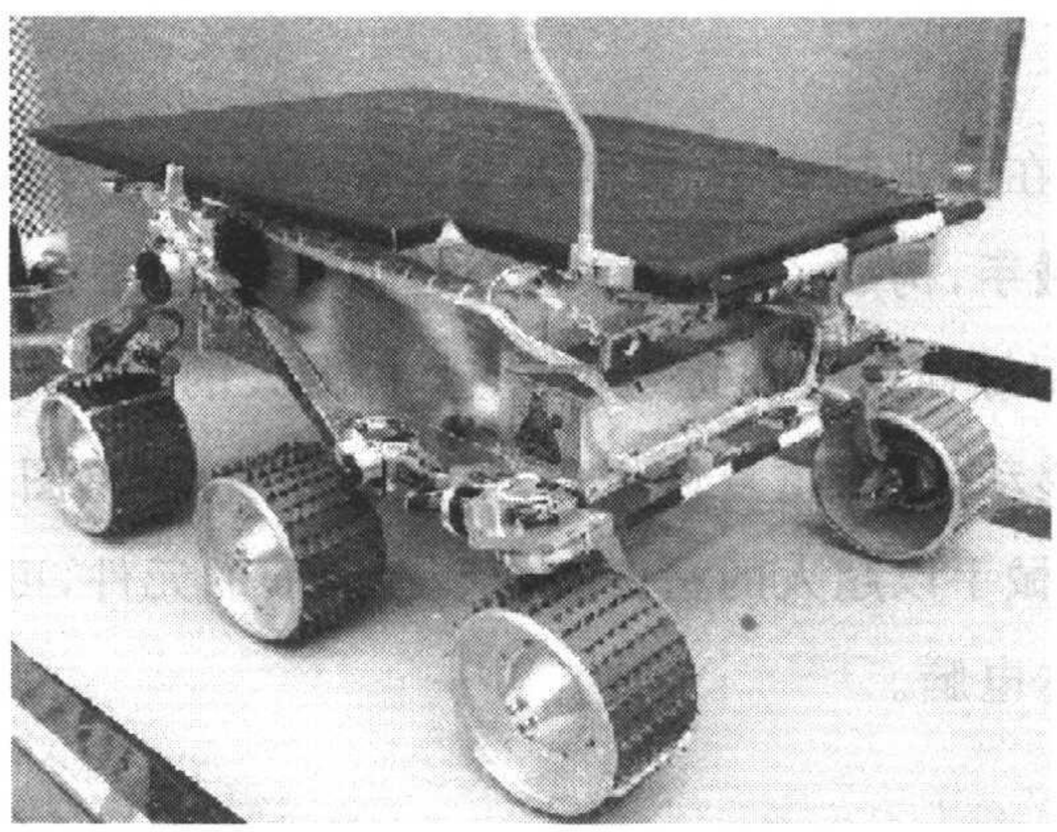


图 1.2 1997 年夏季，在探险者探索金星期间所使用的 Sojourner(旅居者)机器人。它几乎完全由地面遥控。但某些机载传感器考虑了障碍检测(http://ranier.oact.hq.nasa.gov/telerobotics_page/telerobotics.shtm)。© NASA/JPL

在危险和荒凉的环境中，甚至在地球上，这种远距离操纵系统已经得到普及(图 1.3～1.6)。在这些情况下，低复杂度的机器人常常使人类操作员不可能直接控制它的运动。人执行定位和认知活动，但依靠机器人的控制方案提供运动控制。

例如，Plustech 的行走机器人提供腿的自动协调，而人类操作员选择行走的总体方向(图 1.3)。图 1.6 描述的是水下车辆，它控制

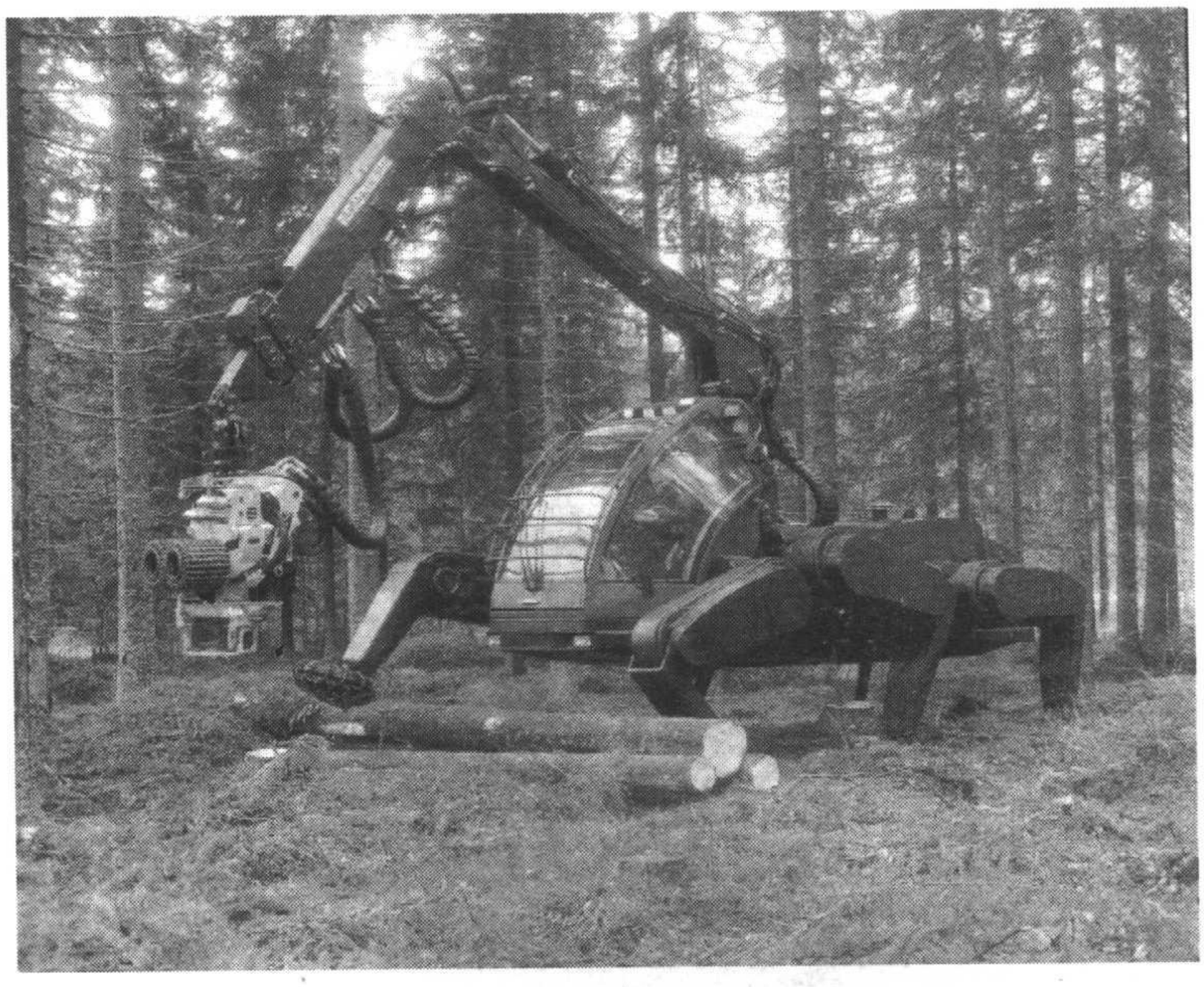


图 1.3 Plushtech 开发的第一个应用驱动的行走机器人。它专门移动砍伐森林的木材。其腿的协调是自动的，但导航依旧由机器人上的人类操作员进行(http://www.plustech.fi)。© Plustech

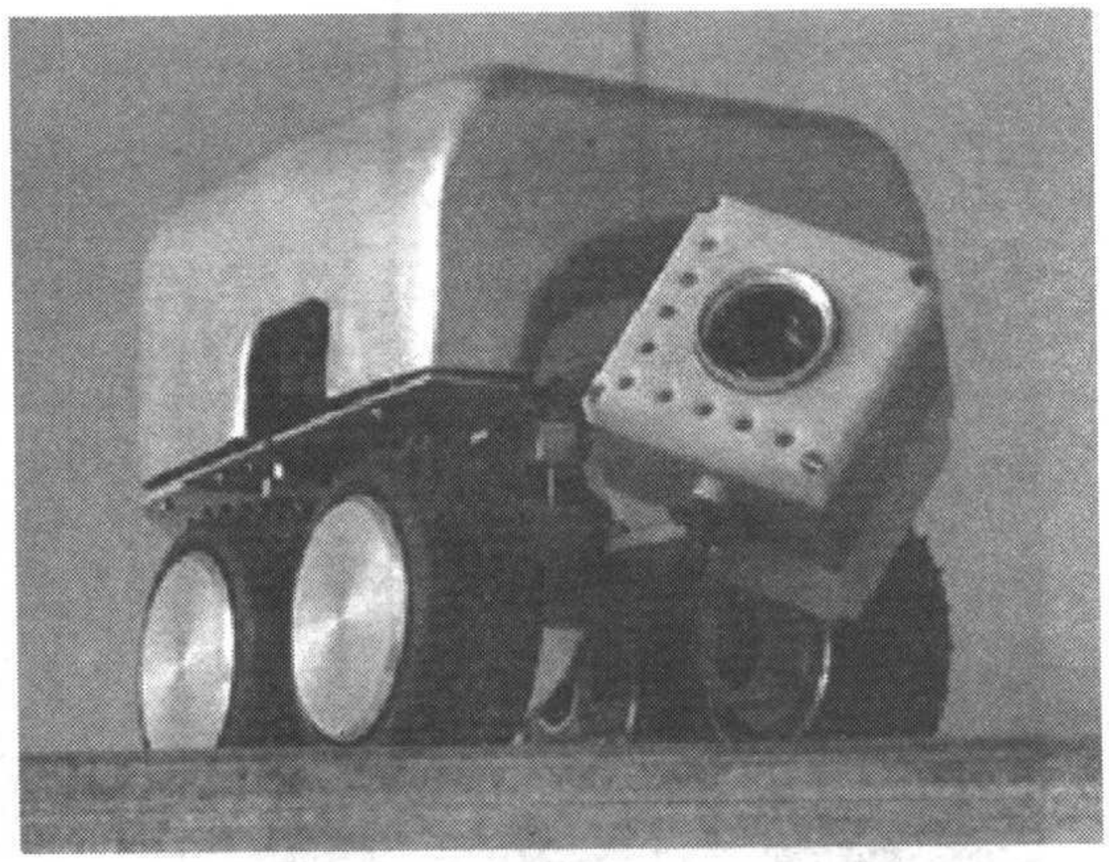


图 1.4 通风管道监测机器人。它以带有变焦距和传感器的全倾斜的摄像机为特征，用作倾斜控制、墙壁跟踪、和交叉口检测(http://asl.epfl.ch)。© Sedirep/EPFL

图 1.5 先锋号图片，在切尔诺贝利专门探索石棺的机器人。© Wide World Photos

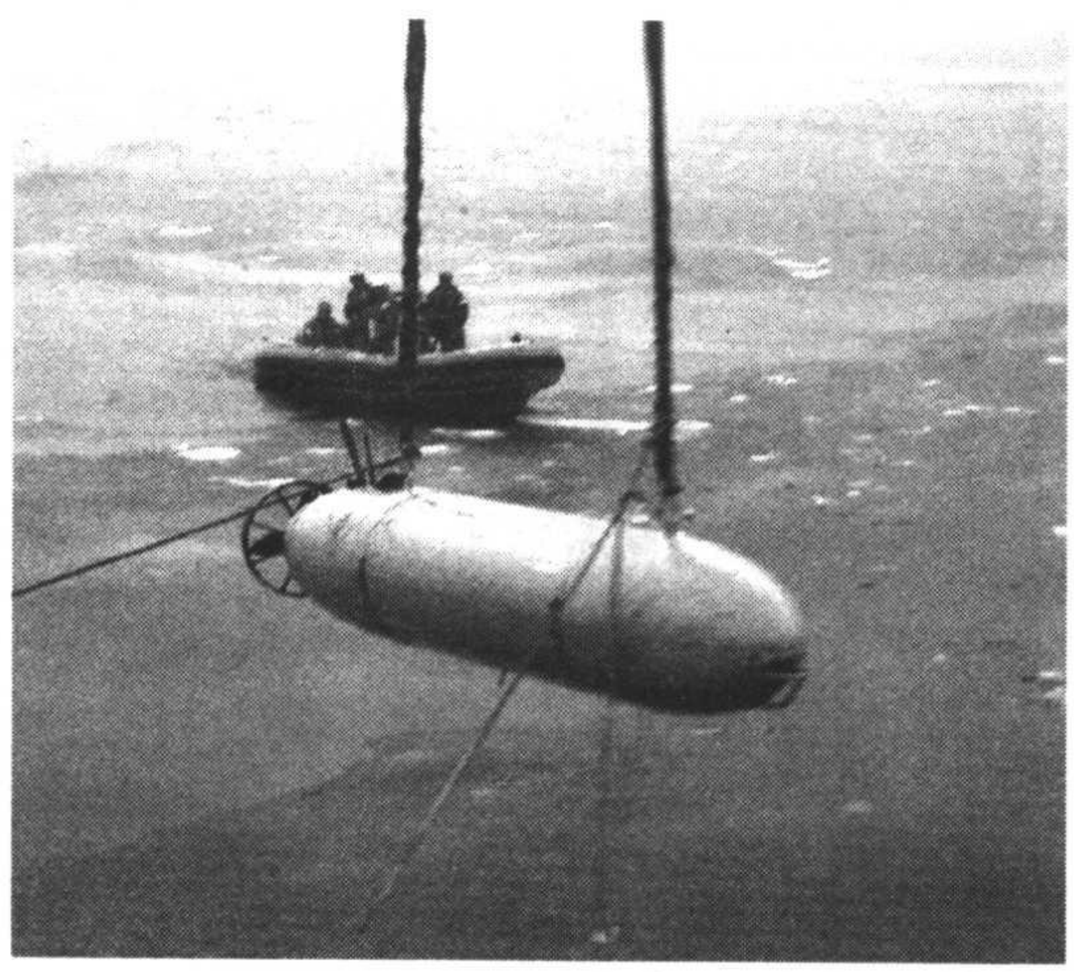


图 1.6 图片：MBARI's ALTEX AUV(自主水下车辆)回收到破冰船 Healy，接着潜入北极冰下。Todd Walsh © 2001 MBARI

6 个推进器，不管水下湍流和水流如何，都能自主地稳定机器人潜水艇，而操作员要做的则是选择潜水艇要达到的目标位置。

其他商业机器人不是在人类不能去的地方运行，而是在人类环境中与人共享空间(图 1.7)。这些机器人之所以引人注目不是因为它们的机动性，而是因为**自主性**。所以，它们在无人干预的情况下，保持对位置的感觉和导航能力是极为重要的。

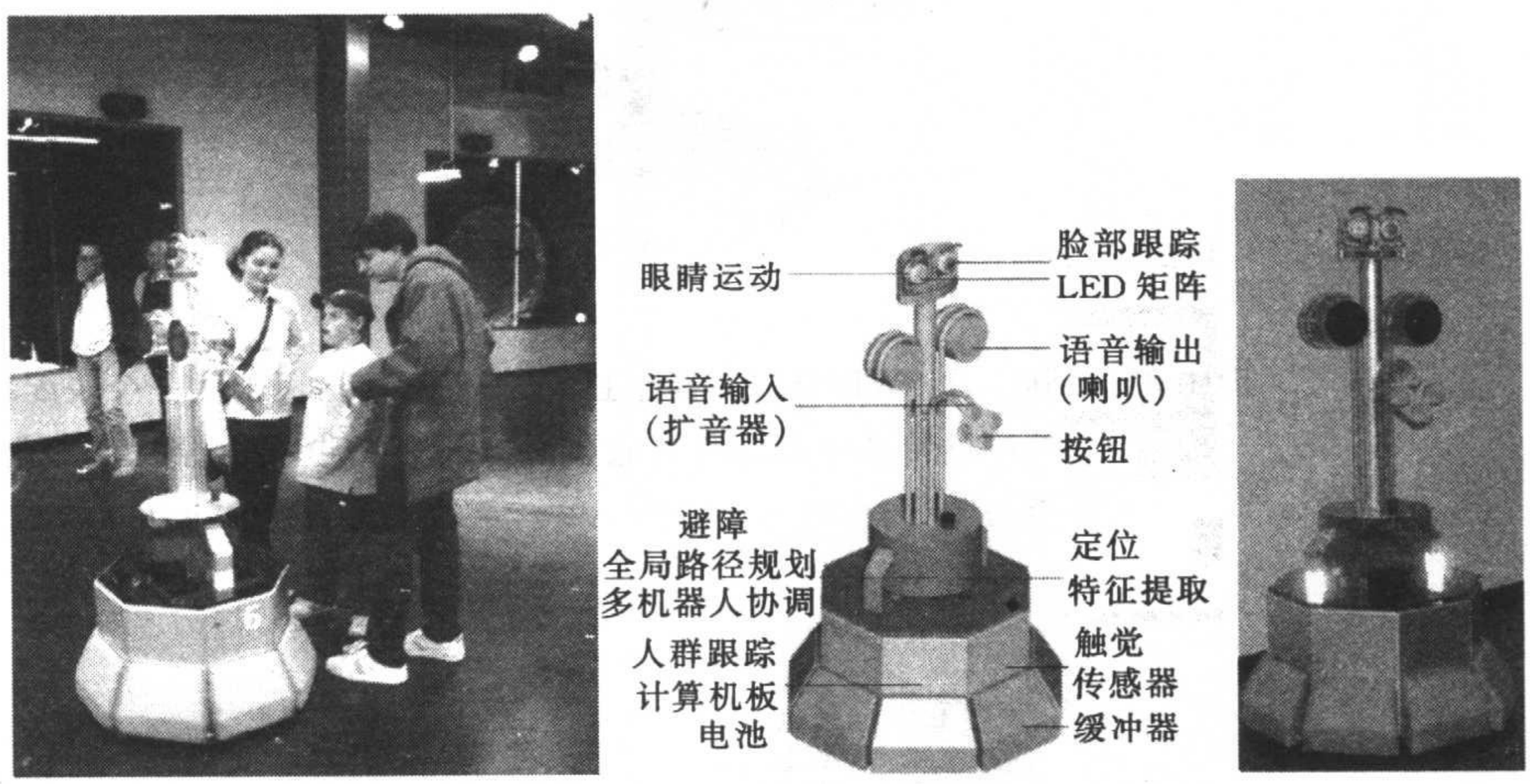


图 1.7 导游机器人能够以教育的方式在展览会上做介绍和交流[48,118,132,143]。10 个机器人已经在为期 5 个月的瑞士展览会 EXPO.02 上会见无数参观者。机器人由 EPFL[132](http://robotics.epfl.ch)开发，由 BlueBotics(http://www.bluebotics.ch)商品化

例如，AGV(autonomous guided vehicle，自主导向车)机器人(图 1.8)，可跟踪用定制传感器做成的专用导向电线，自主地在不同装配站之间分发零件。

Helpmate 服务机器人跟踪天花板上灯的位置走遍医院，传送食物和药品，天花板的灯被人为地指向机器人的前方(图 1.9)。好几个公司已开发了自主清洁机器人，主要用于大型楼宇(图 1.10)。一种这样的机器人正用在巴黎地下铁道。其他专门的清洁机器人利用超市过道正规的几何图式，促进定位和导航任务。

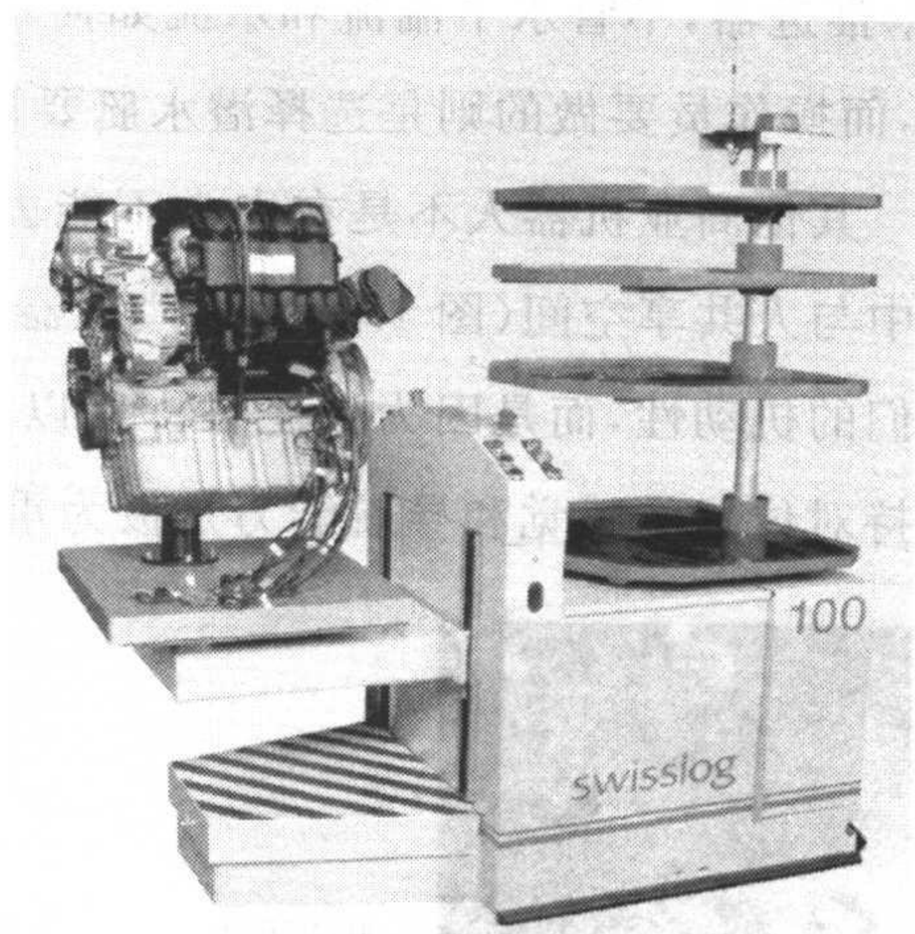


图 1.8 最新一代 SWISSLOG 的自主导向车(AGV),用于从一个装配站到另一装配站传送电机部件。它受装在地板的电线引导。在工业、仓库甚至医院中,由成千上万的 AGV 传送产品。© Swisslog

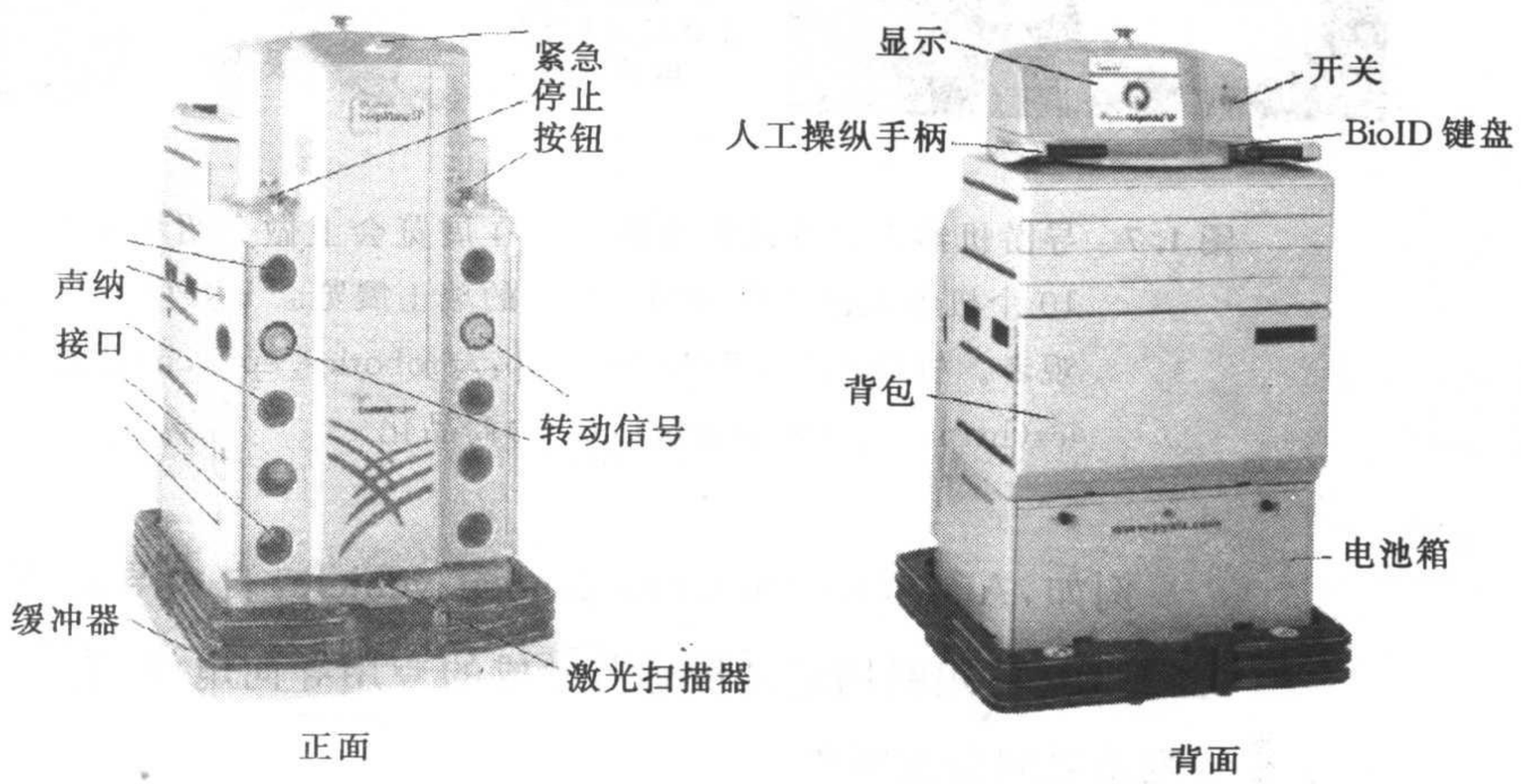


图 1.9 HELPMATE 是用于医院传送任务的移动机器人。它有各种走廊中自主导航、机装的传感器,定位的主传感器是朝向天花板看的摄像机。它可以以天花板的灯或路标作参考(http://www.pyxis.com)。© Pyxis Corp

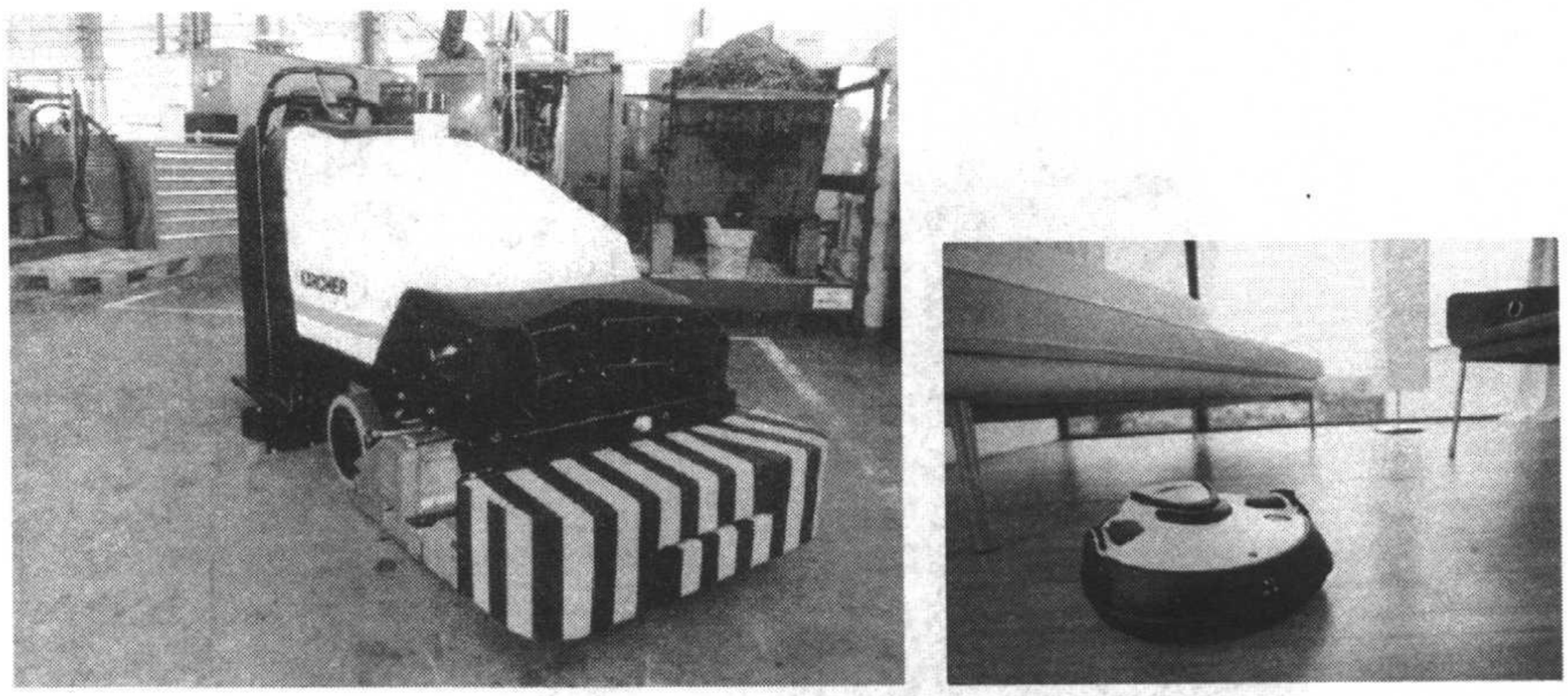


图 1.10 由德国 Alfred Karcher GmbH & Co. 公司开发和销售的 BR 700 工业清扫机器人(左边)和 RoboCleaner RC 3000 家用机器人。BR 700 的导航系统是基于非常复杂的声纳系统和一个陀螺。RoboCleaner RC 3000 具有特定的驱动策略,它覆盖从严重污染区域直到真正干净区域。光学传感器用来测量所吸空气的污染程度(http://www.karcher.de)。© Alfred karcher GmbG & Co

利用标准的、调整为实验室环境服务的机器人研究平台,可以进行认知、定位和导航等高层次问题的深入研究。这是移动机器人当今最大的市场之一。根据尺寸和地形可能性分类,现在有各种编程移动机器人平台。最普遍研究的机器人是那些 ActivMedia Robotics、K - Team SA 和 IRobot(图 1.11～1.13),以及小型机器人,如 EPFL(洛桑的瑞士联邦理工大学)的 Alice(图 1.14)。

虽然,如上所述,移动机器人有应用和市场的广阔范畴,但真正非常成功的移动机器人有一个千真万确的事实:它的设计包含了许多不同知识体的集成。了不起的是,这使移动机器人学达到了成为一个交叉学科领域的程度。为了解决运动问题,移动机器人专家必须了解机械机构、运动学、动力学和控制理论。为了建立鲁棒的感知系统,移动机器人专家必须沟通信号分析领域和专门的知识体系,如计算机视觉等,以便适当地使用众多的传感器工艺技术。定位和导航方面需要计算机算法、信息论、人工智能和概率论知识。

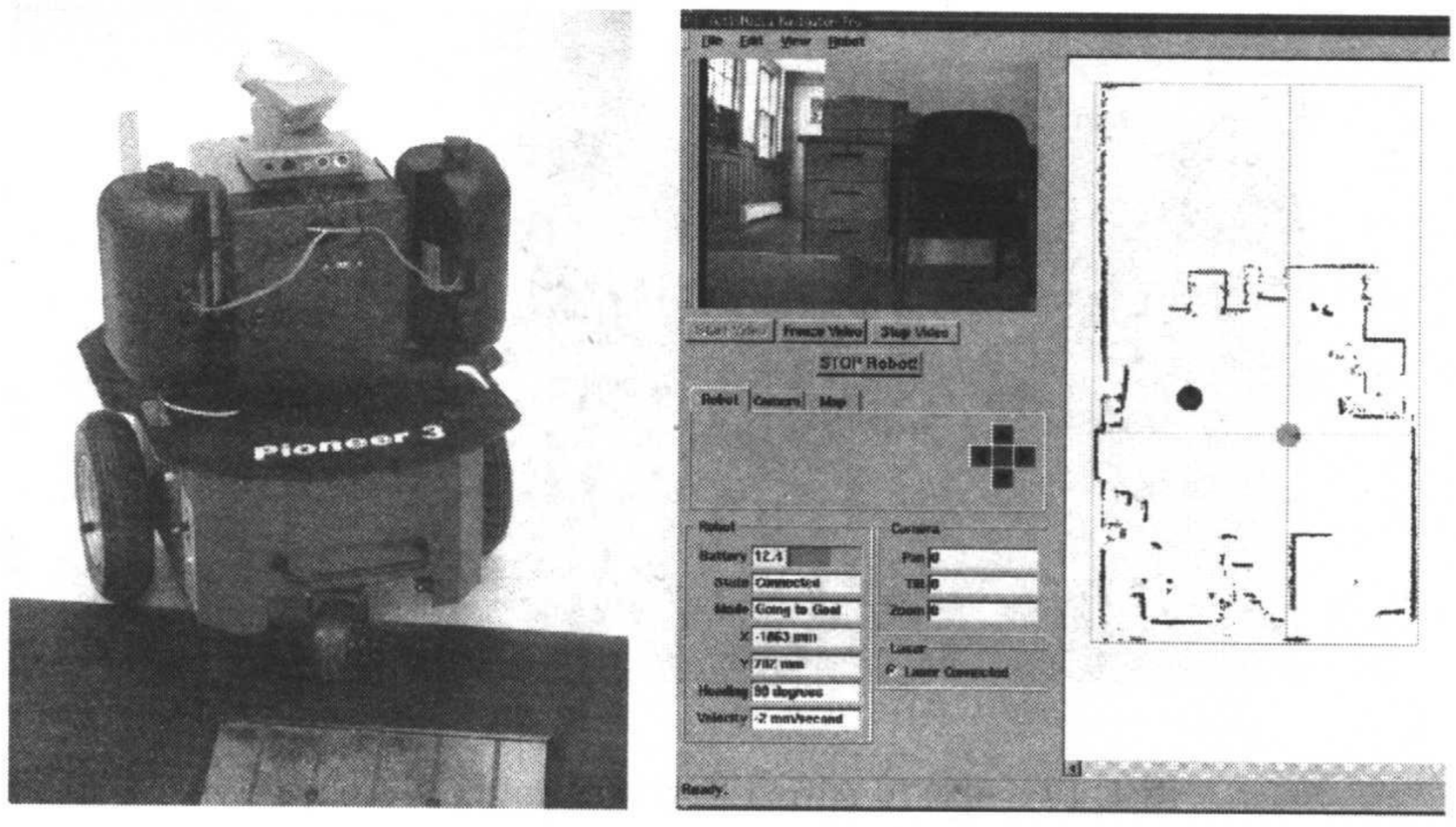


图 1.11 PIONEER(先锋)是一个模块移动机器人。它提供各种选择,像钳子或机装的摄像机等。它装备了由加利福尼亚斯坦福 SRI 开发的精密复杂的导航数据库(复印得到 ActivMedia Robotics 允许,http://www.MobileRobots.com)

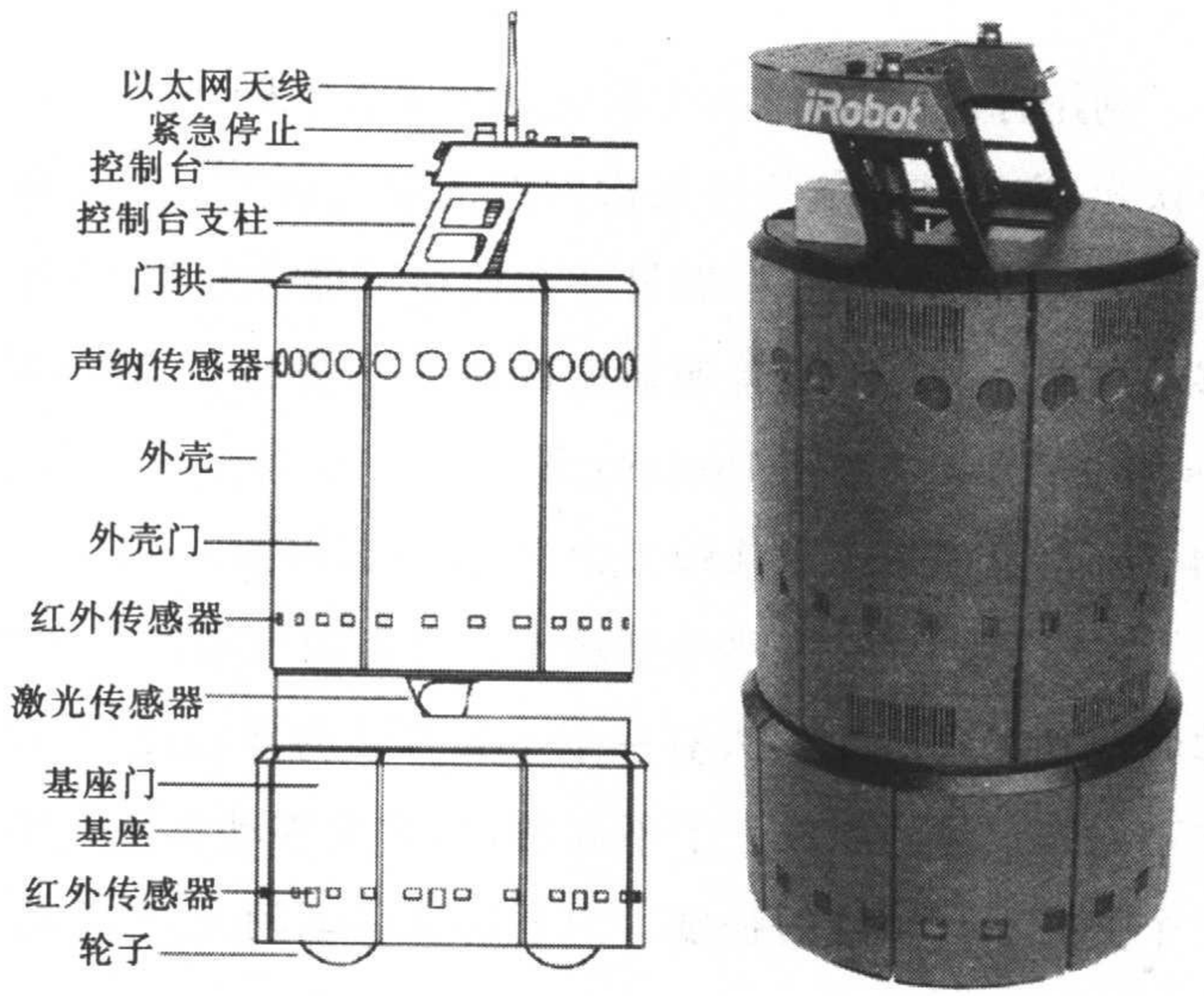


图 1.12 iRobot 的 B21 是一个精密复杂移动机器人,机装 3 个英特尔奔腾处理器。它具有为高性能导航任务而配备的众多类型的传感器(http://www.irobot.com/rwi/)。© iRobot Inc

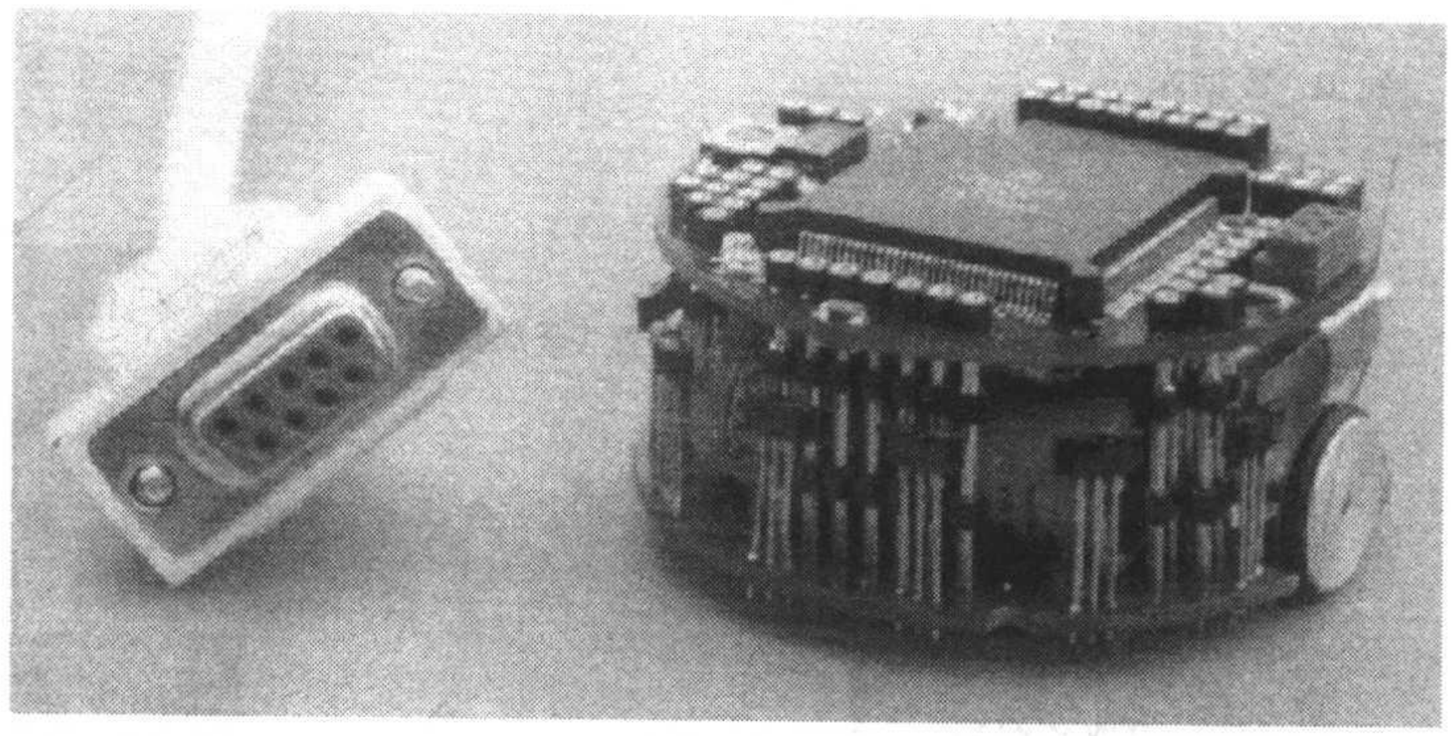
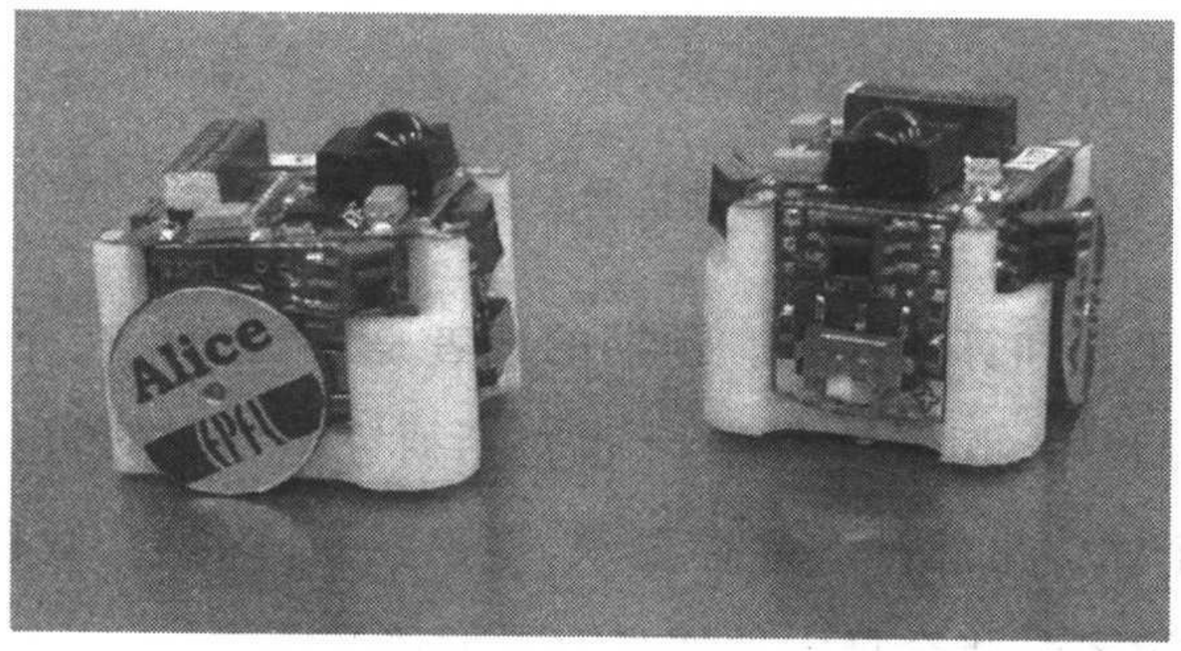


图 1.13 KHEPERA 是研究和教育用的小型机器人，直径大约只有 60 mm。它有各种附加模块，如钳子或摄像机等。到 1998 年年底已经销售了 700 台以上。KHEPERA 由瑞士 K - Team SA 制造和分销（http://www.k—team.com）。© K - Team SA



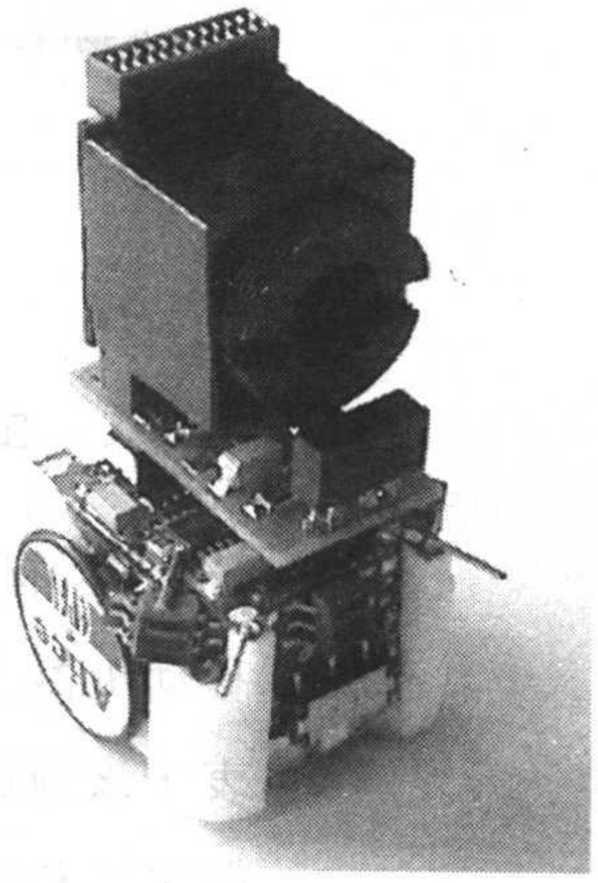


图 1.14 Alice 是最小的全自主机器人之一，它体积约为 2 cm×2 cm×2 cm，它具有约 8 h 的里程表，并使用红外测距传感器、触觉触须线，甚至有导航的小摄像机[54]

图 1.15 表示了贯穿本书要用到的移动机器人系统的抽象控制方案。该图确认了与移动机器人相关的许多知识主体。

本书对移动机器人学的所有方面作了介绍，包括软件和硬件设

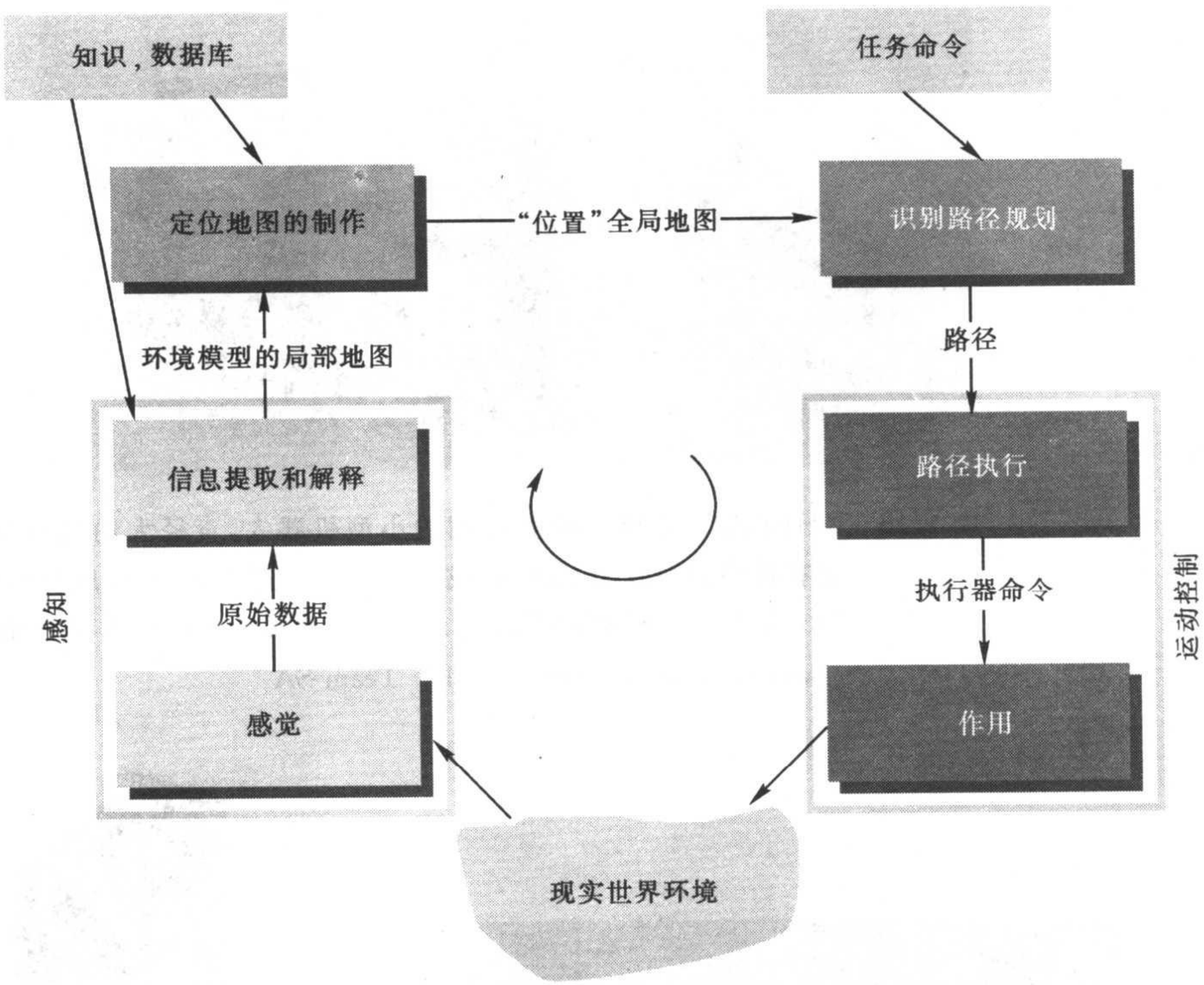


图 1.15 贯穿本书所用的移动机器人参考控制方案

计考虑、有关的工艺和算法技术。预期的读者是广泛的。在介绍性移动机器人学课程中，既包括本科生，也包括研究生，同样还包括热衷于该领域的个体。虽然没有硬性要求，但熟悉矩阵代数、微积分、概率论和计算机程序，将会极大地增强读者的接受能力。

移动机器人学是一个很大的领域，本书并不着重于一般性机器人学，也不把重点放在移动机器人的应用上，而是放在移动性本身。从机制和感知到定位和导航，本书重点放在能使之鲁棒**移动**的技术和工艺学上。

显然，一个有用的、商业上可行的移动机器人完成的不只是移动。它可以用在擦亮超市地板，在工厂守卫，打扫高尔夫球场，在博物馆做导游，或在超市提供导购。有抱负的移动机器人工作者将从

本书开始，但很快地会针对期望的应用，过渡到课程、论文和研究，把从完全不同领域来的技术，诸如人-机（机器人）交互、计算机视觉和语言理解集成起来。

1.2 本书综述

本书介绍机器人模块的不同方面，模块很像图 1.15 所示。第 2 章和第 3 章着重于机器人低层次的**运动能力**。第 4 章对**感知**提出深刻的观点。然后，第 5 和第 6 章我们将跨入较高层次的**定位**难题，以及更高层次的**认知**。特别是，**鲁棒导航能力**。各章建立在前一章的基础上，所以鼓励读者从头开始学习，即使他们的兴趣主要是在高层次上。机器人学的特别之处在于：高层次难题的解决只有在坚实了解系统低层次细节的背景下才有重要的意义。

第 2 章“运动”，以使之运动的最普通机构——轮子和腿的综述开始。无数的机器人例子展示了各运动形式的特殊能力。但是，设计一个机器人的运动系统，要求有定量评估它全部运动功能的能力。第 3 章“移动机器人运动学”，将运动学原理用于整个机器人，从各个轮子对运动的贡献开始，逐渐进入到分析各移动机构所赋予的机器人的机动性。

在常规的移动机器人中，最大的一个缺陷无疑是感知：移动机器人可以跨越地球上许多人造表面而行走，但它不能感知附近环境以及人和动物。第 4 章“感知”，通过提出描述移动机器人性能的清晰语言，开始讨论这个挑战性难题。利用这个现成的语言，第 4 章向移动机器人工作者提出许多不用定制的可用的传感器，描述它们操作的基本原理以及它们性能指标的限制。移动机器人未来最有希望的传感器是视觉，第 4 章包括和综述了电荷耦合器件（CCD）和互补金属氧化半导体（CMOS）传感器两者的工作原理和限制。

但是，感知不止是感觉，感知也是用有意义的方法对被感知数据的**解释**。第4章的后半部分，描述了在移动机器人学应用中一直是最有用的特征提取策略，包括从基于测距感知数据提取几何式样，以及利用基于视觉感知的路标和全图像分析。

用运动机构装备并配有感知的硬件和软件，移动机器人可以运动和感知环境。首先，机动性和感觉必须满足的是它的定位：移动机器人需要经常保持对位置的感知。第5章“移动机器人定位”，描述了不需直接定位的方法。然后，研究成功的定位策略的基本组成部分：信任度表示方法和地图表示方法。实例研究展示了不同的定位方案，包括马尔可夫定位和卡尔曼滤波器定位。第5章的最后部分专门讨论了挑战性问题，以及移动机器人自主地描绘它们周围环境最有希望的技术。

移动机器人是一个如此年轻的学科，以至缺乏标准化的结构体系，而且还没有建立机器人的操作系统。但是，当人们强调移动机器人更高层次的能力时，体系结构问题是极其重要的：机器人如何始终如一地从一处到另一处进行鲁棒导航、解释数据、定位和控制它的运动呢？这个机器人的最高水平的能力，我们称为**导航能力**，有许多显示特别结构策略的移动机器人。第6章“规划和导航”，综述了机器人导航的现状，指出当前不同的技术是十分相似的，主要差别在于机器人控制问题的**分解**方式。但是，第6章首先强调，有能力的导航机器人通常必须展示的两个技巧：避障和路径规划。

关于移动机器人的交叉学科领域，需要知道的远远比本书所包含的多得多。虽然我们希望，本书广泛的介绍会将读者置于移动机器人学集体智慧的背景中，但这仅仅是开始。幸运的话，你编程或制作的第一个机器人会对你只说赞许的话。

第 2 章　运动

2.1　引言

移动机器人需要运动机构，它能够使机器人在它的环境中无约束地运动。但是运动有众多不同的可能途径，因此机器人运动方法的选择是移动机器人设计的一个重要方面。在实验室中，已经有能够行走、跳跃、跑动、滑动、溜冰、游泳、飞翔和滚动的研究机器人。这些运动机构中的大部分一直受到它们生物学上的对应物的启示（图 2.1）。

然而有一个例外：有源动力轮是人类发明的，它在平地上达到极高的效率。这一机构对生物系统而言，不完全是外来的。我们的两足行走系统可由一个滚动的、边长为 d（相当于步伐跨距）的多边形来近似（图 2.2）。随着步距的减小，多边形就逼近为一个圆或轮。但是在自然界中，没有创造出完全旋转、有源动力的关节，而这正是轮式运动所需的技术。

生物系统在穿越各类崎岖不平环境的运动中取得成功，因此有希望模拟它们对运动机构的选择。但是，由于若干原因，在这方面复制自然界是很困难的。首先，在生物系统中，通过结构的复制可以容易地实现机械的复杂性。结合专业化知识，细胞分割技术可以很容易地制造出一个有几百条腿和成千上万根感知纤毛的千足虫。在人

运动类型	运动阻力	运动的基本运动学
渠中的流动	液体动力	漩涡运动
爬行	摩擦力	纵向振动
滑行	摩擦力	横向振动
跑动	动能消耗	多节摆的摆动
跳跃	动能消耗	多节摆的摆动
行走	万有引力	多边形的滚动（图 2.2）

图 2.1 生物系统中用到的运动装置

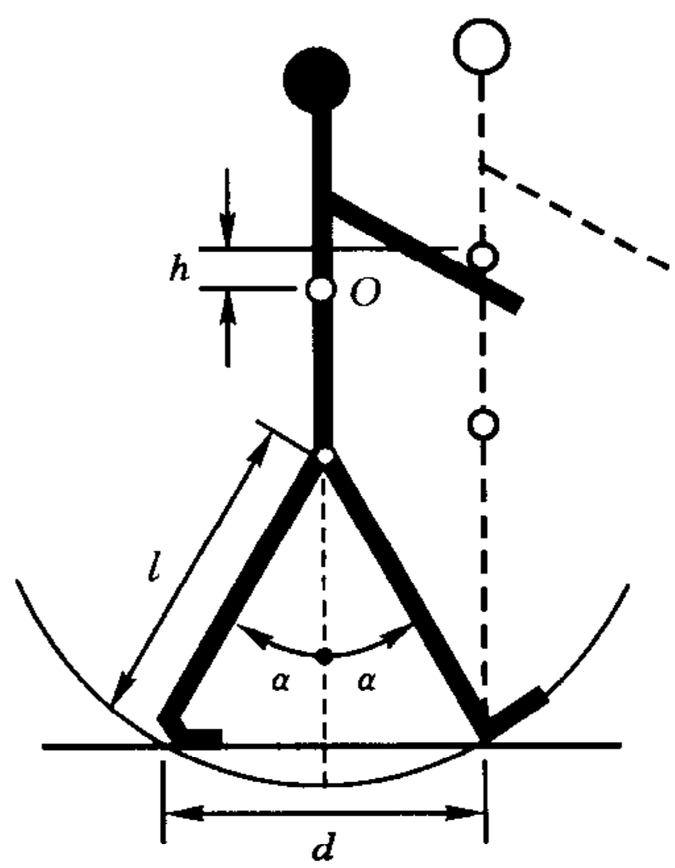


图 2.2 一个两足的行走系统近似于一个滚动的多边形，多边形的边长为 d，等于步伐的跨距。随着步距的减小，多边形接近于半径为 l 的圆或轮子

造结构中，各部分必须单个地制作，因此不存在这种规模的机体。此外，细胞是能极小化的微观组件，由于微小的尺寸和重量，昆虫达到了人类制造技术还一直不能做到的鲁棒性水平。最后，大型动物和昆虫所用的生物能量存储系统、肌肉及液压激励系统，其产生的转矩、响应时间和转换效率远远超过了同等规模的人造系统。

由于这些限制因素，一般移动机器人运动，要么使用轮式的机构——一种众所周知的车辆人造工艺技术；要么使用数目不多的有关节的腿——一种生物运动方法的最简单形式(图 2.2)。

一般来说，和轮式运动相比，腿式运动要求更高的自由度，因此有更大的机械复杂性。除了简单以外，轮子非常适合于平地。如图 2.3 所示，在平坦的表面，轮式运动比腿式运动效率高出 1～2 个数

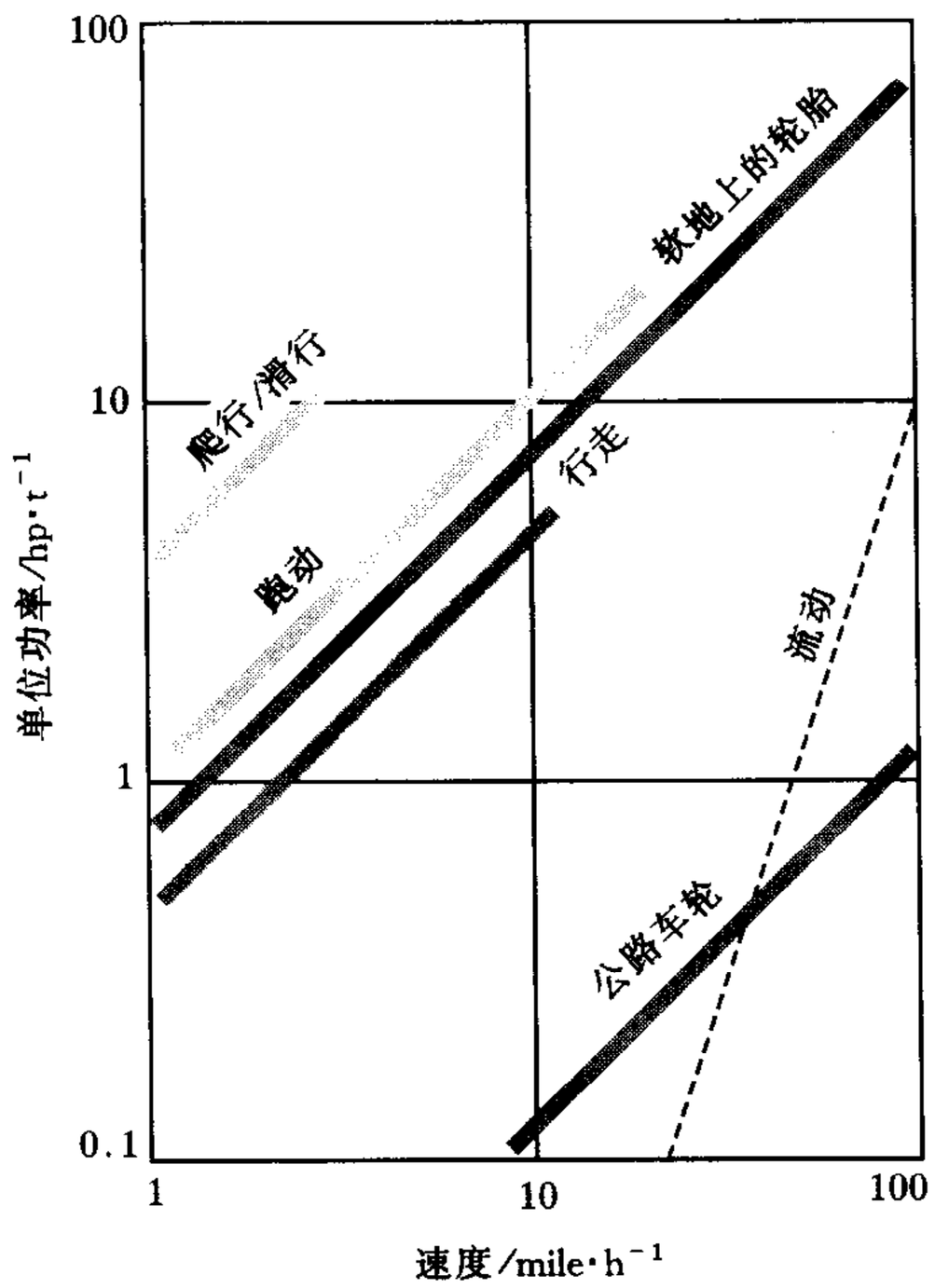


图 2.3 特定功率和各种运动机构所能获得的速度的对照[33]
(注：1 英里(mile)＝1609.344m　1 英马力(hp)＝745.700W)

量级。铁道就是一种理想的轮式运动的工程，因为坚硬和平坦的钢

铁表面使滚动摩擦减至极小。但是，如果表面很软，轮式运动因滚动摩擦而变得无效，而腿式运动受摩擦较小，因为它与地面只是点接触。在图2.3中，展示了轮胎在软地面情况下效率急剧损失。

实际上，轮式运动的效率主要依赖于周围环境的质量，尤其是地面的平整度与硬度；而腿式运动的效率则依赖于腿和身体的质量，在步行过程中的所有点上，机器人必须支撑这两者。

可以理解，自然界偏爱腿式运动，因为运动系统实际上必须在粗糙和非结构地形中运行。例如，森林中的昆虫，地面高度的垂直变化常常比昆虫总高度大一个数量级。出于同样的原因，人类环境，不论是室内还是室外，经常由修建过的平整表面组成。因此，也可以理解，所有的移动机器人的工业应用实质上都利用了某种形式的轮式运动。最近，对较自然的室外环境，在面向混合和腿式工业机器人方面已获得某些进展，如图2.4所示的树林机器人。

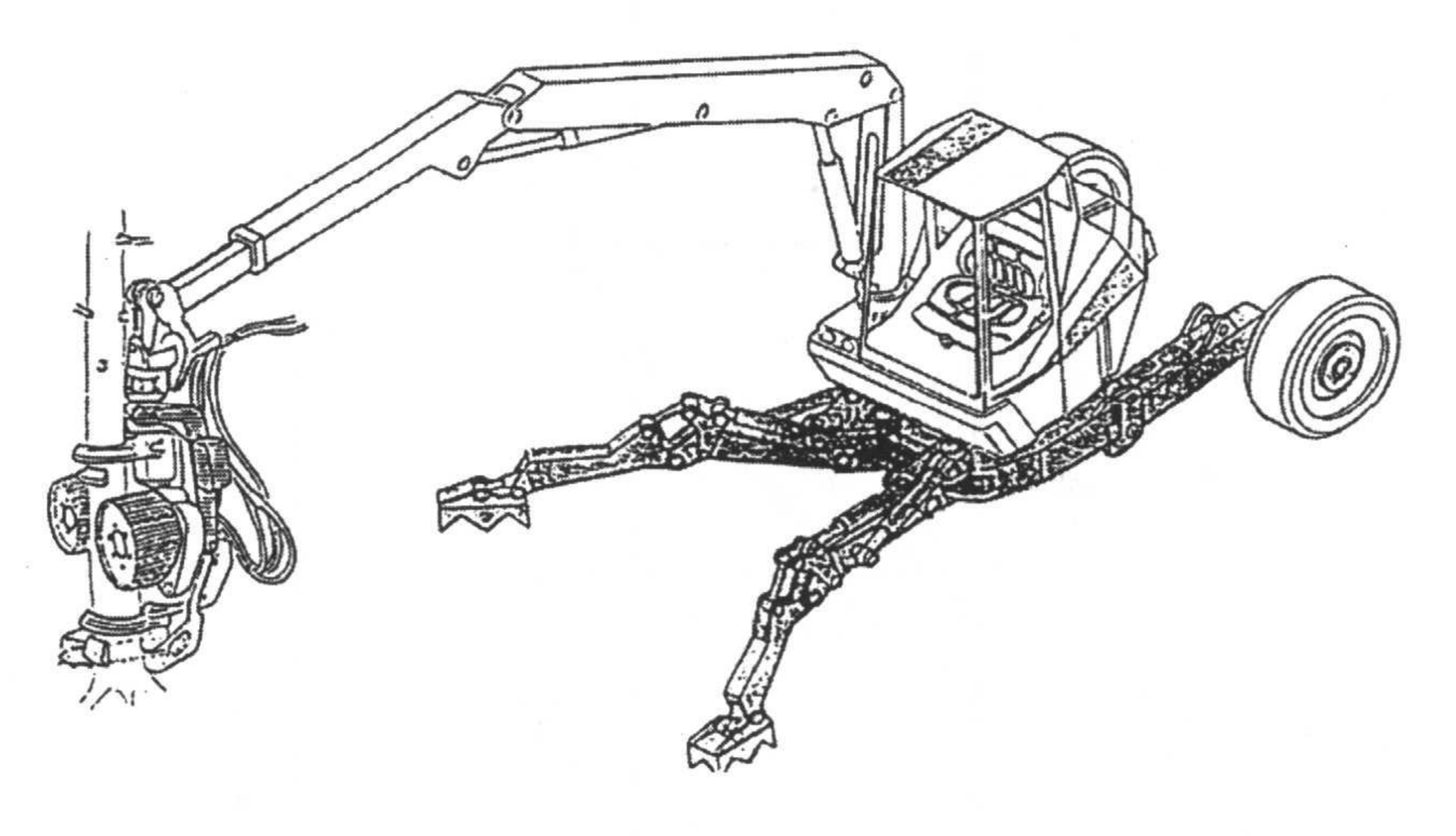


图2.4 RoboTrac，用于粗糙地形的腿-轮混合的车辆[130]

在2.1.1节，我们提出有关移动机器人运动的所有方面的一般考虑。接着，在2.2和2.3节中我们将综述移动机器人腿式运动和轮式运动的技术。

2.1.1 运动的关键问题

运动是操纵的补充。在操纵中，机器人手臂是一定的，但通过施力，它们在车间里做搬运工作；周边环境是固定的，机器人对环境施力以完成运动。在这两种情况下，科学的基础就是研究产生相互作用力的执行器和实现期望运动学与动力学特性的机构。因此，运动和操纵共有相同的稳定性、接触特征和环境类型等核心问题：

- 稳定性
  - ——接触点的数目和几何形状
  - ——重心
  - ——静态/动态稳定性
  - ——地形的倾斜
- 接触特征
  - ——接触点/路径的尺寸和形状
  - ——接触角度
  - ——摩擦
- 环境类型
  - ——结构
  - ——媒介（如水、空气、软或硬的地面）

对运动的理论分析可从机械学和物理学开始。从这点出发，我们可以正式地定义和分析移动机器人运动系统的所有方式。但是，本书重点在于移动机器人的导航问题，特别强调感知、定位和认知。因此，我们不深入钻研运动的物理基础。不过，这个章节的其余两节给出了腿式运动[33]和轮式运动的综述。然后，在第 3 章提出了轮式移动机器人运动学和控制的更详细分析。

2.2 腿式移动机器人

腿式运动以一系列机器人和地面之间的点接触为特征。其主要优点包括在粗糙地形上的自适应性和机动性。因为只需要一组点接触，所以只要机器人能够保持适当的地面步距，这些点之间的地面质量是无关紧要的。另外，只要行走机器人的步距大于洞穴的宽度，它就能跨越洞穴或者裂口。腿式运动的最后一个优点是，有高度的技巧来操纵环境中的物体。举一个精彩的昆虫例子，即甲壳虫，它用灵巧的前肢在运动的同时能够滚动一个球。

腿式运动的最主要的缺点包括动力和机械的复杂性。腿，可能包括几个自由度，必须能够支撑机器人部分总重量，而且在许多机器人中，腿必须能够抬高和放低机器人。另外，如果腿有足够数目的自由度，在许多不同的方向给予力，机器人就能实现高度的机动性。

2.2.1 腿的构造与稳定性

因为腿式机器人受到生物学上的启发，所以检验一下生物学上成功的有腿系统是有益的。许多不同的腿的构造已经在各种各样的生物体中成功地存在(图2.5)。大型动物，如哺乳动物和爬行动物有

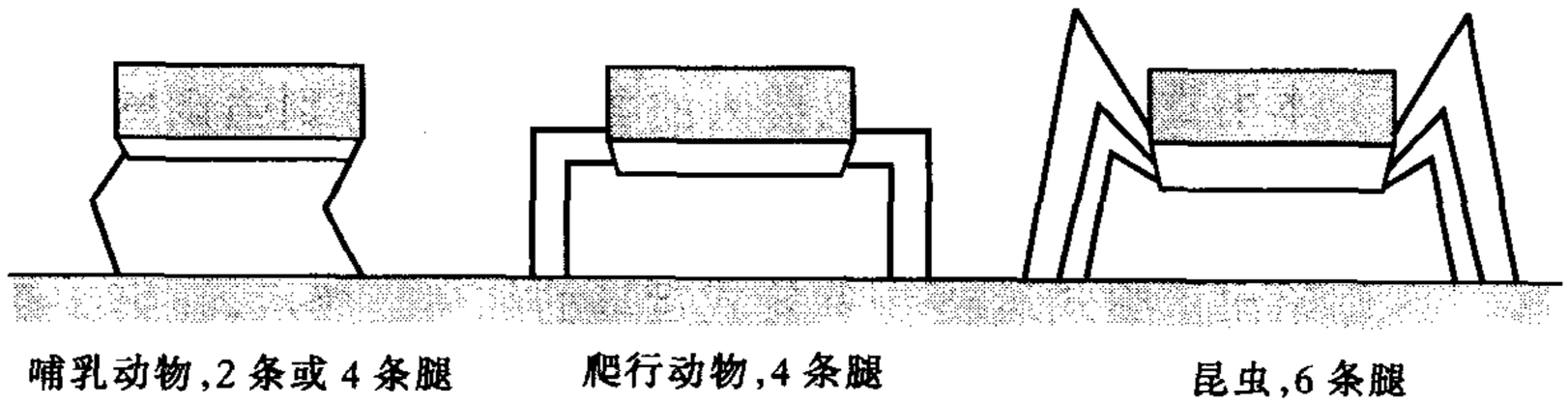


图 2.5 各种动物的腿的配置

4条腿，而昆虫有6条腿或更多。某些哺乳动物，仅靠2条腿行走的能力已经很完美。尤其是人类，平衡能力已经进展到甚至可用单腿

进行跳跃的水平*。这种异常的机动性是以很高代价得来的：保持平衡的更复杂的主动控制。

相反，3 条腿的动物假定它能保证其重心处在地面接触的三脚区内，它就能够展示静止、稳定的姿态。如 3 条腿的凳子所展现那样，静止稳定性意味着不需要运动而保持平衡。在没有外力时，稳定性小的偏离（比如轻轻推凳子）会被动地予以校正而趋向稳定的姿态。

但是，机器人必须能够抬腿行走。为了能达到静态行走，机器人必须至少有 6 条腿。在这种结构下，有可能设计出一种步态。按此，腿的静态稳定三脚区总是与地面接触（图 2.8）。

昆虫和蜘蛛一出生立即能行走。对它们来说，行走时的平衡问题比较简单。4 条腿的哺乳动物不能稳定地行走，但是用 4 条腿能很容易地站立。比如野生动物，在它们能站起来之前要花几分钟来尝试，然后又要花好几分钟学习行走而不摔倒。2 条腿的人类，甚至不能以静止稳定方式站在一个位置。婴儿需要几个月才能站立和行走，甚至需要更长时间来学习跳跃、跑步和单腿站立。

各个腿的复杂性种类繁多，也存在潜在的能力。再者，生物世界提供了丰富的处于两个极端的例子。例如，毛虫利用液压，通过构建体腔和增加压力使各腿伸展，而且通过释放液压使各腿纵向地回收，然后刺激单个可拉伸的肌肉，牵引腿靠向身体。各条腿只有 1 个自由度，它沿着腿纵向地定方向。前向运动依赖于体内的液压，它能伸张两腿间的距离。毛虫的腿在机械上很简单，即利用最少数目的外表肌肉，完成了复杂的整体运动。

在另一极端，连同脚趾的深层刺激，人腿有 7 个以上的主自由度，15 个以上的肌肉群，激励 8 个复合的关节。

在腿式移动机器人情况下，通常要求至少 2 个自由度，通过提起腿和将腿摆动向前，使腿向前运动。更普通的是对更复杂的移动，如

* 在儿童的生长过程中，用于确定孩子是否获得高级运动技能的测试之一是单腿跳跃能力。

图 2.6 所示，附加了第 3 个自由度。在创造两足行走机器人中，最新产品已在踝关节处加了第 4 个自由度。通过刺激脚底板的姿态，足踝能做到更一致地与地面接触。

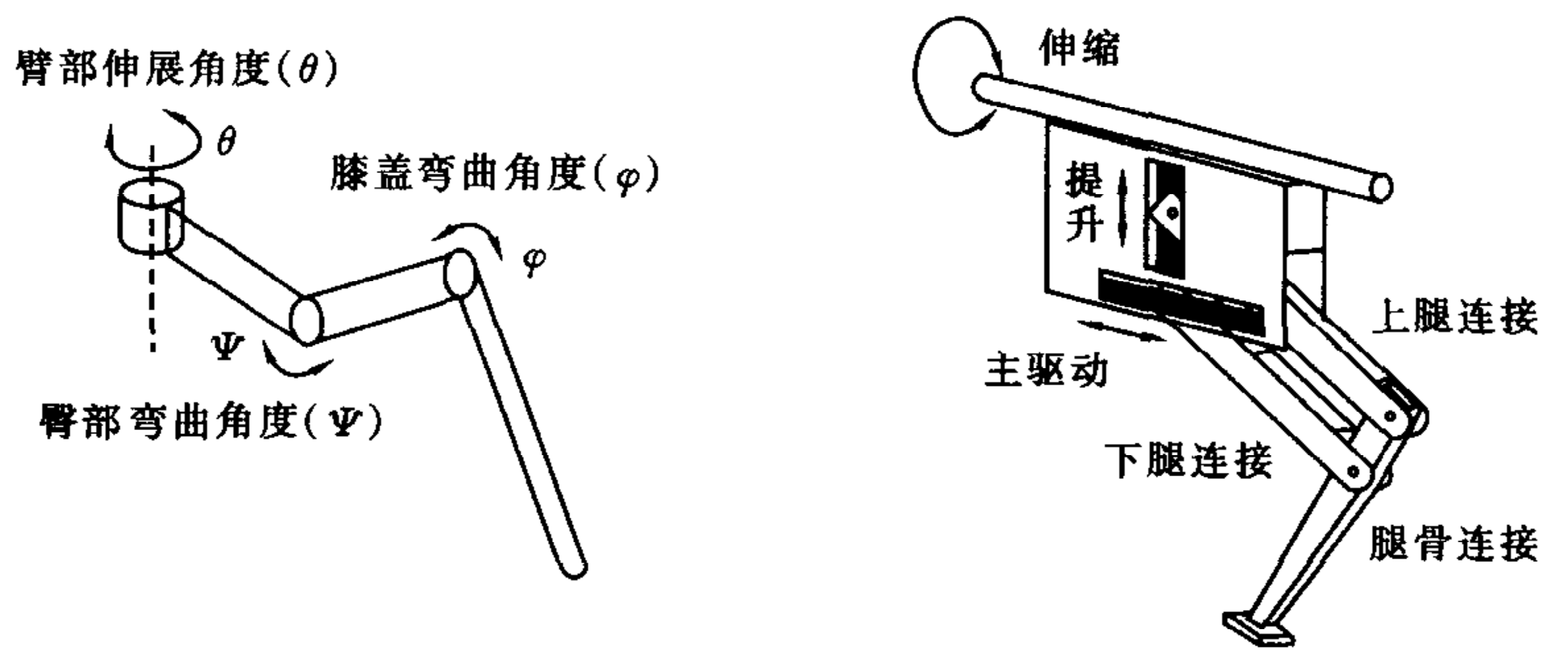


图 2.6 具有 3 个自由度腿的两个例子

总之，增加机器人腿的自由度提高了机器人的机动性，既扩大了机器人能行走的地形范围，又增强了机器人以各种步态行走的能力。当然，附加关节和激励器的主要缺点是带来动力、控制和质量方面的问题。附加的激励器需要能量和控制，它们都被加到腿的质量上，进一步增加了对现有激励器的能量和负荷的要求。

对于多腿移动机器人，存在运动时腿的协调或步态控制问题。可能的步态数目依赖于腿的数目[33]。对单条腿而言，步态是抬起与放下事件的序列。对一个有 k 条腿的移动机器人，步行机器可能事件的总数 N 为

$$N = (2k-1)! \tag{2.1}$$

对于 2 条腿的步行器，$k=2$ 腿，可能事件的总数 N 为

$$N = (2k-1)! = 3! = 3 \cdot 2 \cdot 1 = 6 \tag{2.2}$$

6 个不同事件为：

1. 抬右腿；
2. 抬左腿；
3. 放右腿；

4. 放左腿；

5. 2 条腿一起抬起；

6. 2 条腿一起放下。

当然，该事件迅速地增长得很大。例如，一个 6 条腿的机器人，理论上步态远远超过：

$$N = 11! = 39916800 \tag{2.3}$$

图 2.7 与 2.8 描述了几个 4 腿的步态和静态的 6 腿三脚架式的步态。

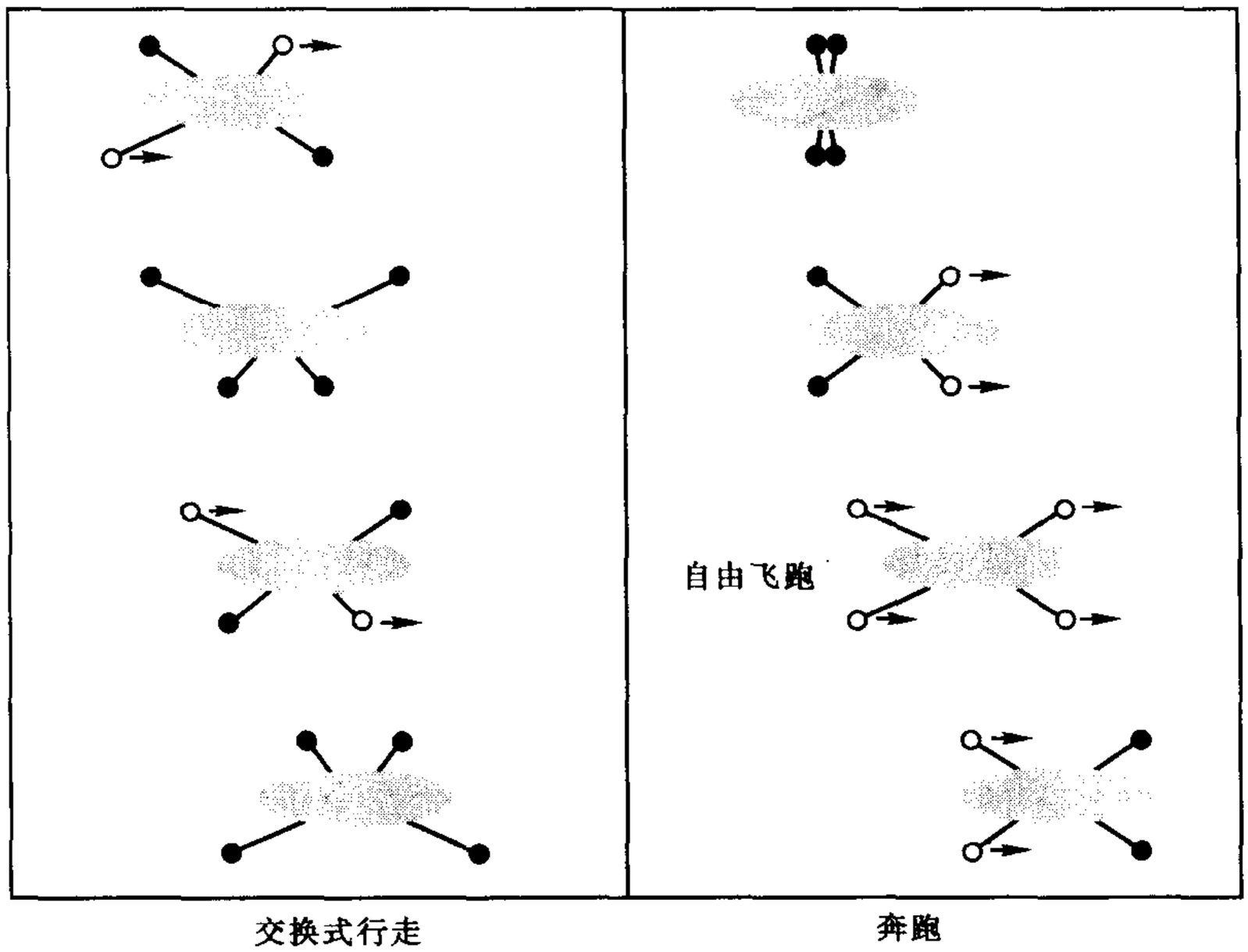


图 2.7 4 条腿的两种步态。因为这种机器人少于 6 条腿，因此它不可能静态行走

2.2.2 腿式机器人运动的例子

尽管迄今为止还没有大量的工业应用，但是腿式运动仍是一个长期研究的重要领域。下面给出几个令人感兴趣的设计，从 1 条腿的机器人开始并以 6 条腿机器人结束。有关对爬行和行走机器人的一个很好的综述，请浏览 http://www.uwe.ac.uk/clawar/。

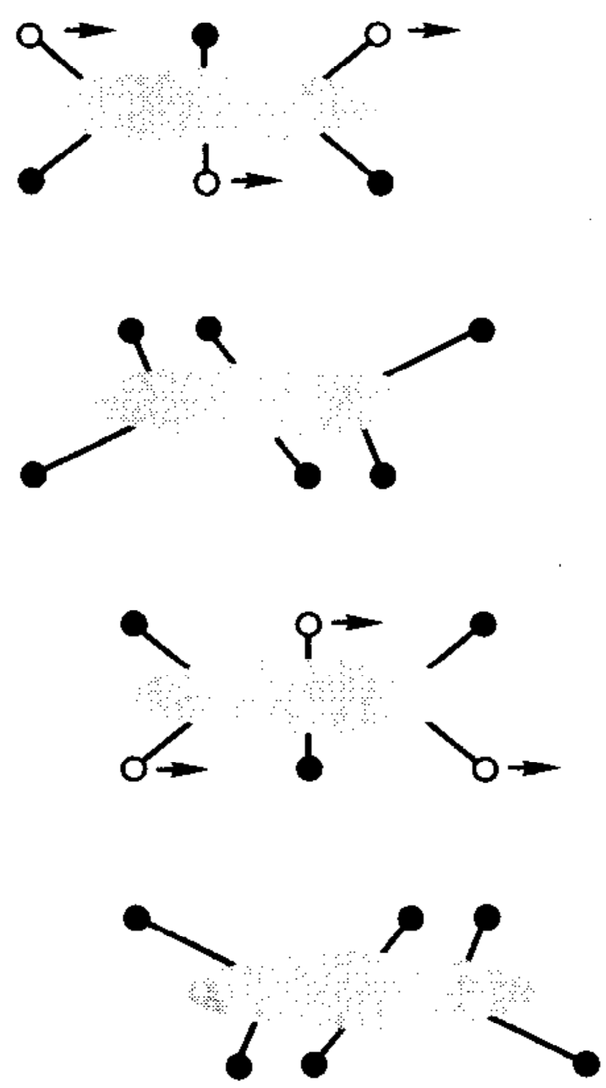


图 2.8 6 条腿的静态行走。3 条腿形成的三脚架一直存在

2.2.2.1 单腿机器人

腿式机器人腿的数目最少当然是 1。有几个理由说明，使腿的数目尽量少是有益的。机身的质量对行走机器是极其重要的，而 1 条腿使累加的腿的质量最小。当机器人有几条腿时，就要求腿之间的协调，1 条腿就不需要这种协调了。也许最重要的是，单腿机器人使腿式运动的基本优点最大化：在全跟踪的场合，如轮子一样，腿与地面只有一个接触点。单腿机器人只需要一系列的单点接触，就能经受最粗糙的地形。而且，取一个跑步的起点，跳跃机器人可动态地跨过比它的步幅大的沟隙。而多腿不能跑的行走机器人，只限于跨过与它的步幅一样大小的沟隙。

制造单腿机器人的主要困难是保持平衡。1 条腿的机器人不仅不可能静态行走，而且在静止时保持静态稳定也不可能。机器人或者改变它的重心，或者给出校正力，总之必须主动地自我平衡。因此，成功的单腿机器人必须动态地稳定。

图 2.9 表示的是 Raibert 跳跃机[28,124]，它是已制作的最有名的

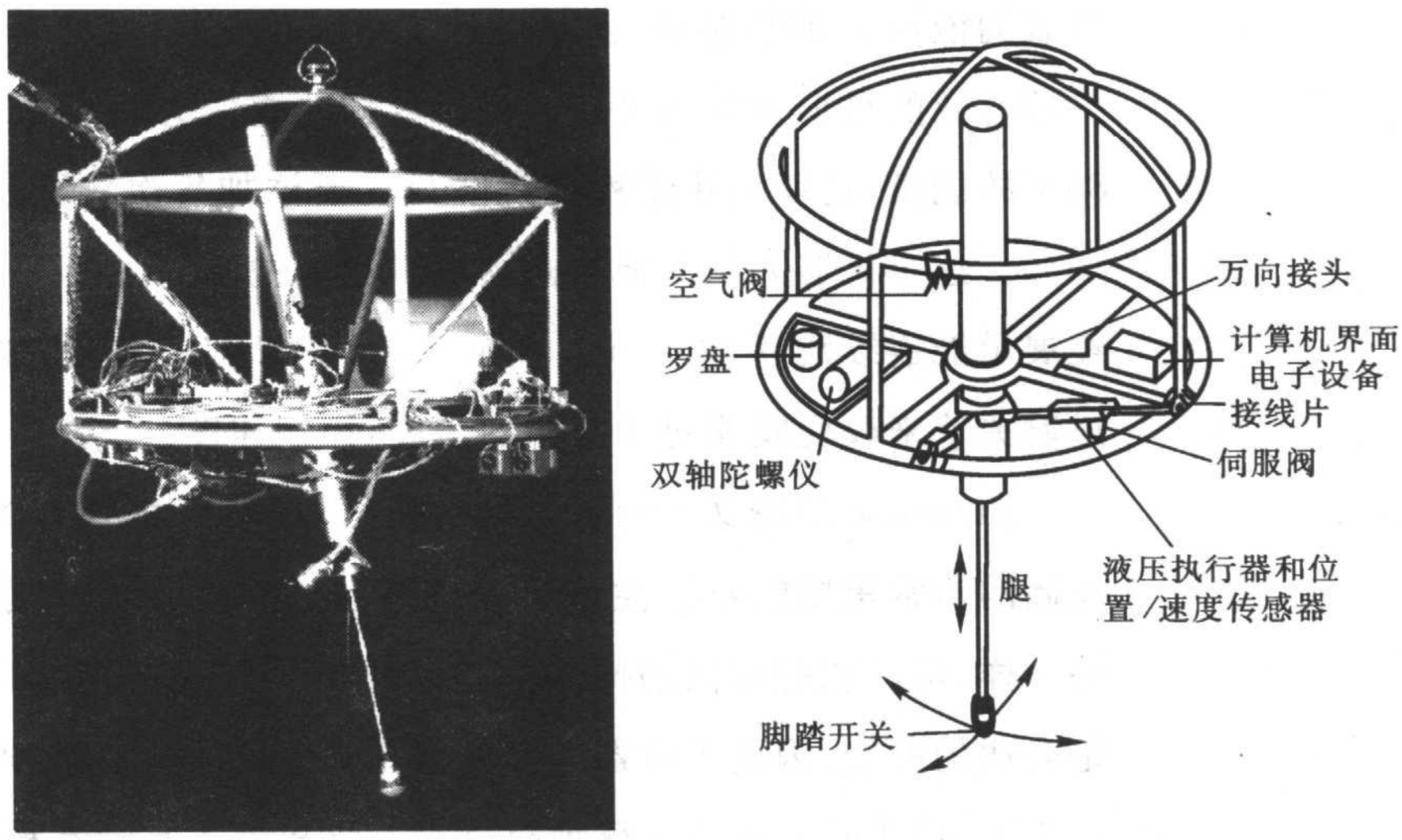


图 2.9 Raibert 跳跃机[28,124]。图片由 LegLab 和 Marc Raibert 提供。© 1983

单腿跳跃机器人之一。这个机器人通过调节腿相对于身体的角度，不断地修正身体姿态和机器人速度。激励是液压的，包括在立姿向空中跳跃期间，腿的高性能纵向伸展。激励虽然是强有力的，但这些激励器需要一个随时连接到机器人的大型的、外装的液压泵。

图 2.10 表示的是最近研制开发的更高效能的结构[46]。它不用

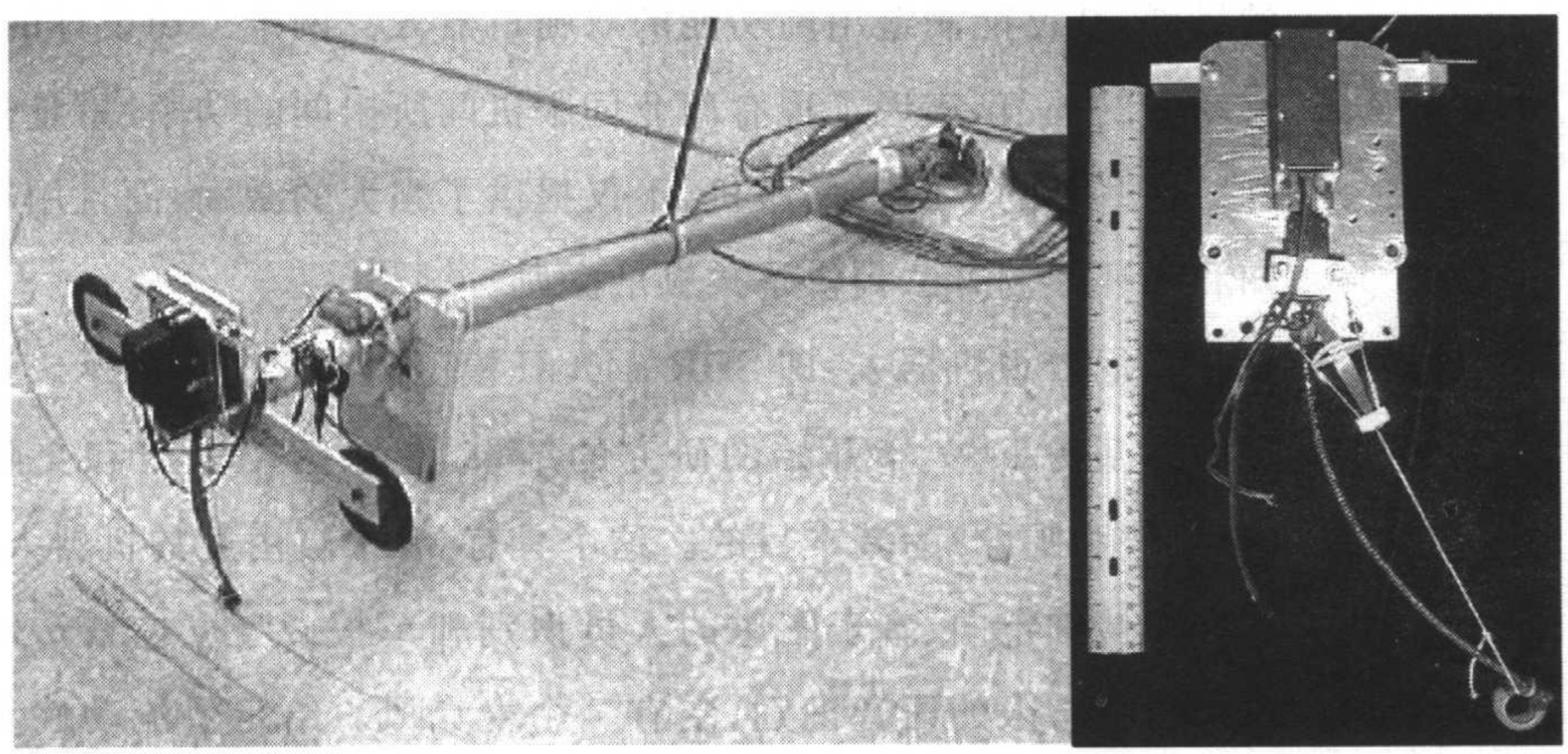


图 2.10 一个简单的二维弓腿跳跃机[46]。图片由 CMU 的 H. Benjamin Brown 和 Garth Zeglin 提供

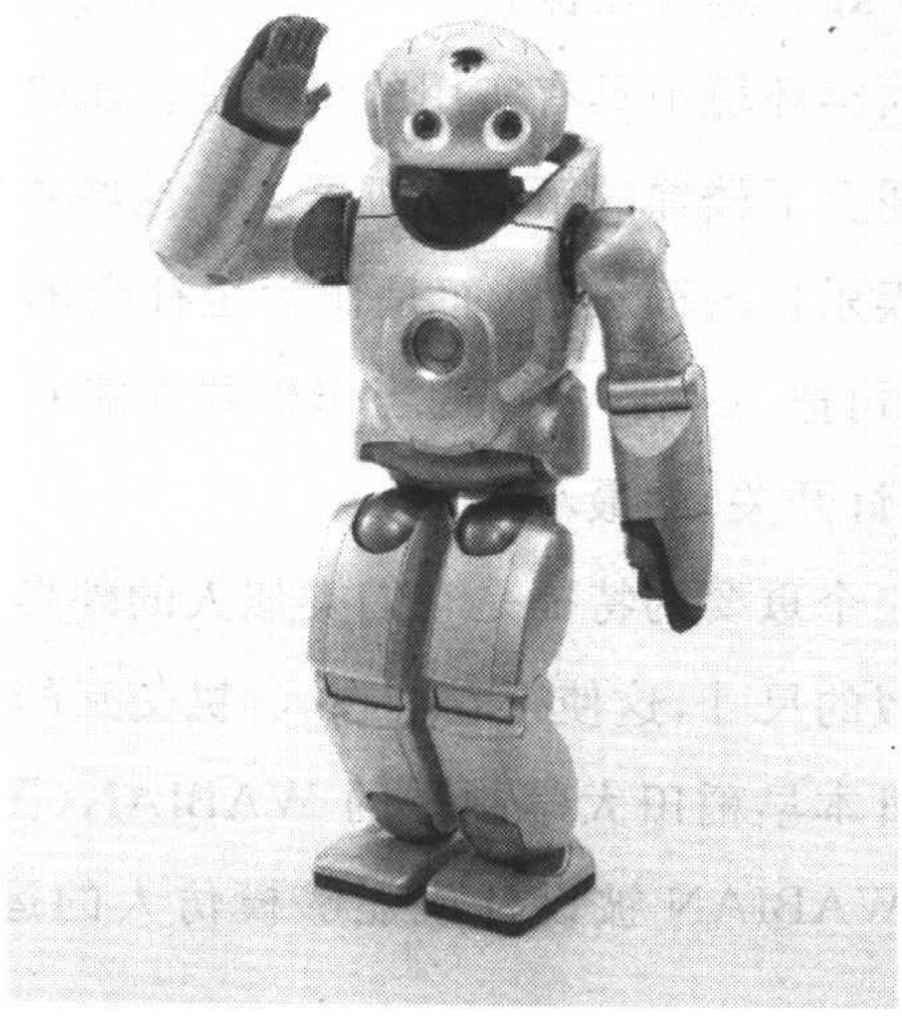
外装的液压泵供给能量，而是设计了弓腿跳跃机，当它着地时，利用一根有效的弓形弹簧腿来获得机器人的动能。这个弹簧返回了几乎85%的能量，这就意味着稳定跳跃只需要添加每次跳跃所需能量的15%。这个机器人用支架轴向固定，用装在机器人上的一组电池，已经展示能连续跳跃20 min。和Raibert跳跃机一样，弓腿跳跃机通过髋关节来改变腿相对于身体的角度，控制速度。

Ringrose的论文[125]揭示了应用于单腿跳跃机的有关机械学和控制的非常重要的对偶性。通常，精巧的机械设计与复杂的控制电路一样，可以执行相同的操作。在该机器人中，其脚的物理形状正好是合适的曲线，因此当机器人非完全垂直着地时，由碰撞产生合适的校正力，使机器人在下次落地时垂直。该机器人是动态稳定的，而且是被动的。机器人与其环境之间的物理交互提供了校正，不用计算机或环路中的任何主动控制。

2.2.2.2 双腿机器人(双足)

在过去的10年中，已经展示了各种类型的成功的双腿机器人。已经证明双腿机器人能跑、跳和上下楼梯行走，甚至玩空中把戏，如翻筋斗。在商业部门，本田和索尼公司在过去10年已经做出了重大的进展，并已能制造出高功能的双腿机器人。两个公司设计的小型、有动力的关节，实现了现有商业伺服机闻所未闻的功率/重量性能指标。这些新的“智能”伺服机，不仅提供了强大的激励，而且用转矩感知和闭环控制方法提供了柔顺的激励。

索尼梦想机器人，即SDR-4X II模型，如图2.11所示。这个流行的模型是1997年开始的研究的结果，它以运动表演和交际娱乐(如跳舞和唱歌)为基本目的。这个具有38个自由度的机器人，为了声音的精细定向、基于图像的人员识别和单板微型立体景深图像的重构以及有限的语音识别，装有7个麦克风。给定了流动性和运动表演的目标，索尼公司花费了巨大的精力设计了标准的应用系统的运动，这能使他们的技术人员能以简捷的方式编排舞蹈。要注意的

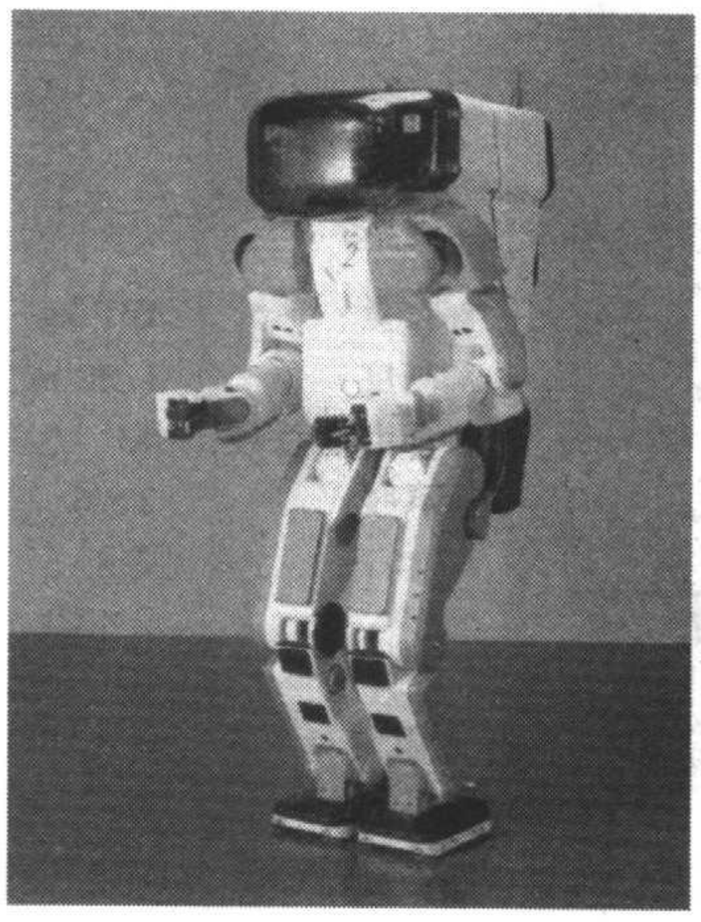


规格：
重量： 7 kg
高度： 58 cm
颈自由度： 4
身体自由度：2
臂自由度： 2×5
腿自由度： 2×6
五指手

图 2.11 索尼 SDR－4XⅡ © 2003 Song Corporation

是，SDR－4X II 比较小，只有 58 cm 高，6.5 kg 重。

本田公司的拟人机器人项目有着值得注意的发展经历。此外，它还抓住了传动方面非常重要的工程挑战。图 2.12 展示了模型



规格：
最大速度：2 km/h
自由度： 15 min
重量： 210 kg
高度： 1.82 m
腿自由度：2×6
臂自由度：2×7

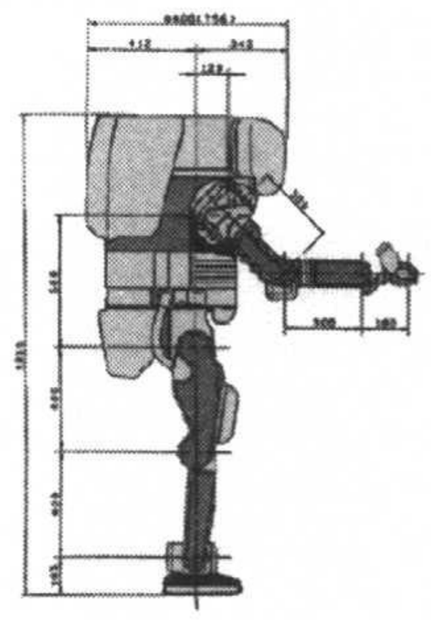
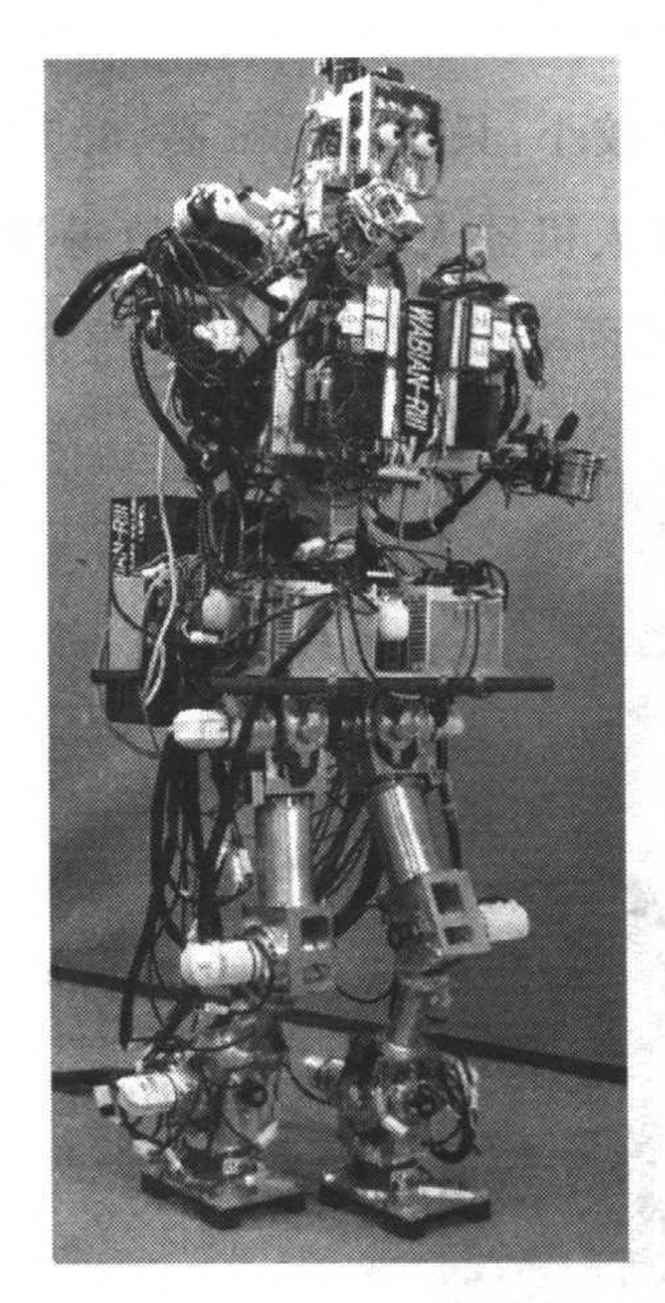


图 2.12 日本本田公司的拟人机器人 P2。© Honda Motor Corporation

P2，它是最新模型 Asimo（在开创移动性方向有超前步伐）最直接的

先驱。由图示可以注意到，本田拟人机器人比 SDR－4X II 大得多，高 182 cm，重 210 kg。这可在保持不危险尺寸和姿势的同时，容许在楼梯和台阶的人类环境中具有实用的机动性。也许这是令人满意地展示仿生的双腿上下楼梯的第一个机器人。这些本田拟人机器人系列不是专门为娱乐目的而设计，而是作为全社会的人类帮手而设计。例如，本田公司把 Asimo 的高度当作机器人能对人类环境进行管理操作（如控制灯开关）的最小高度。

双足机器人一个重要的特征是它们类似人的外形。它们可以被制造成和人相近似的尺寸，这使得它们在人-机交互的研究中成为出色的交通工具。日本早稻田大学制造的 WABIAN（图2.13）正是作这种研究[75]的。WABIAN 被设计成能够模仿人的运动，甚至能够像人一样跳舞。



规格：

重量：	131 kg
高度：	1.88 m
自由度总数：	43
下肢：	2×6
躯干：	3
臂：	2×10
颈：	4
眼睛：	2×2

图 2.13 日本早稻田大学的拟人机器人 WABIAN－RIII[75]。图片由早稻田大学的 Atsuo Takanishi 提供

双足机器人只能在某些限制内静态地稳定。因此，像 P2 和 WABIAN 这样的机器人即使站着不动，通常也必须连续地进行伺服

平衡校正。而且,各条腿都必须有足够的容量以支撑机器人全部重量。在四腿机器人情况下,平衡问题随各条腿上的负荷需求而变得容易解决。双腿机器人的一个一流设计是麻省理工学院设计的Spring Flamingo(图 2.14)。这个机器人与腿激励器串接,插入了弹

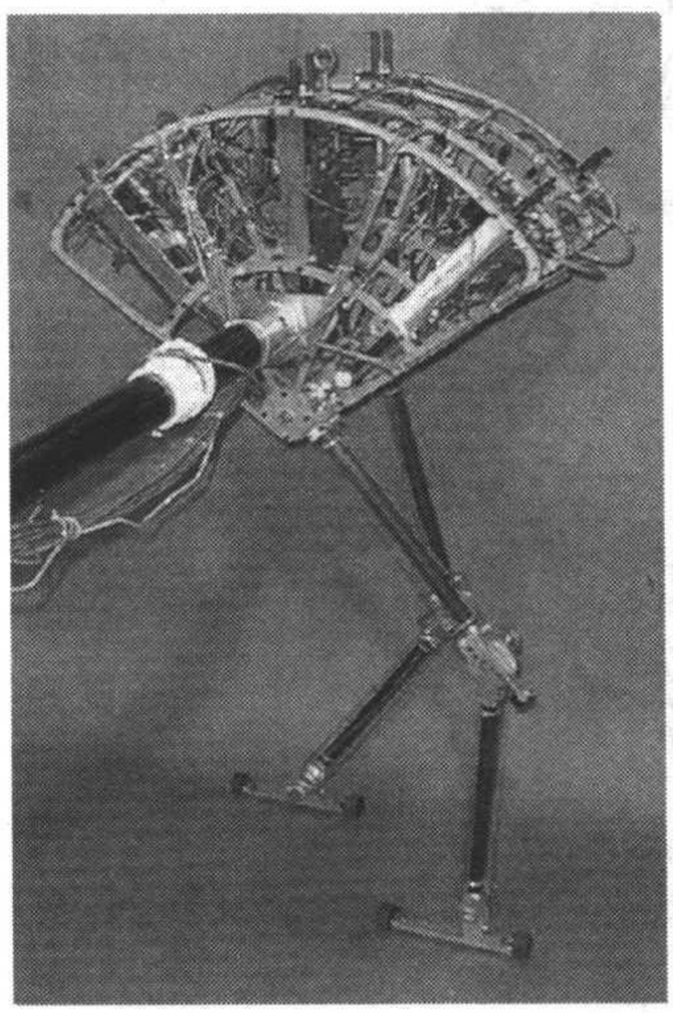
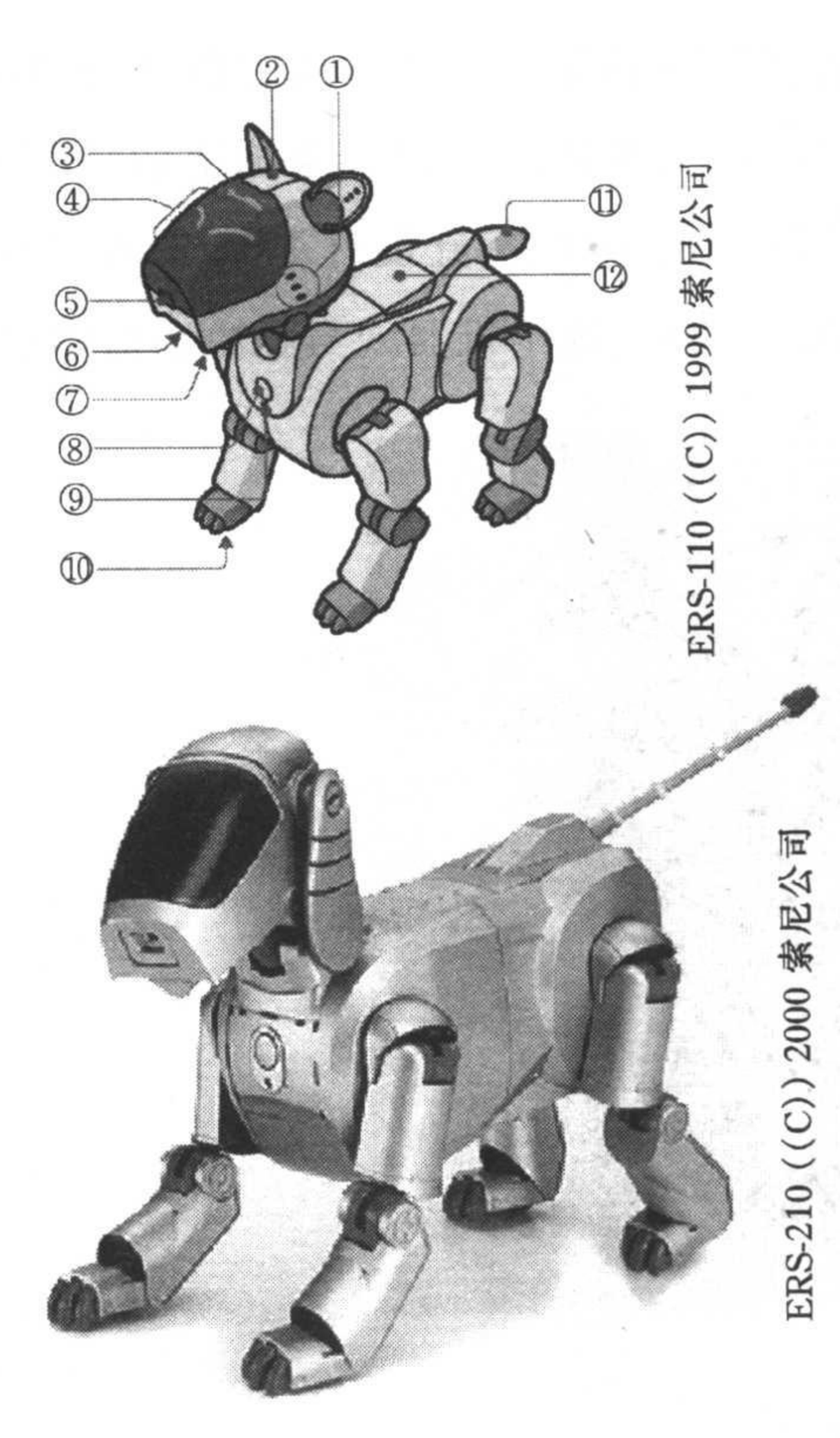


图 2.14 麻省理工学院设计的 Spring Flamingo[123]。图片由 Jerry Pratt,MIT leg Laboratory 提供

簧,实现更有弹性的步态。通过与限制膝盖关节角度的"膝盖骨"结合,Flamingo 实现了惊人的仿生运动。

2.2.2.3 四腿机器人(四足)

尽管四腿站立不动可以被动地稳定住,但要行走仍具有挑战性。因为在步行期间,为了保持稳定,机器人的重心必须主动地偏移。最近,索尼投资数百万美元开发一个叫做 AIBO 的四腿机器人(图 2.15)。为了制作这个机器人,索尼开发了近实时的新的机器人操作系统和新的啮合伺服电机。它具有足够大的力矩以支持机器人,还保证安全的可后向驱动。除了开发专用的电机和软件之外,索尼公司还插入了一个彩色视觉系统,使 AIBO 能够追逐一个色彩鲜艳的

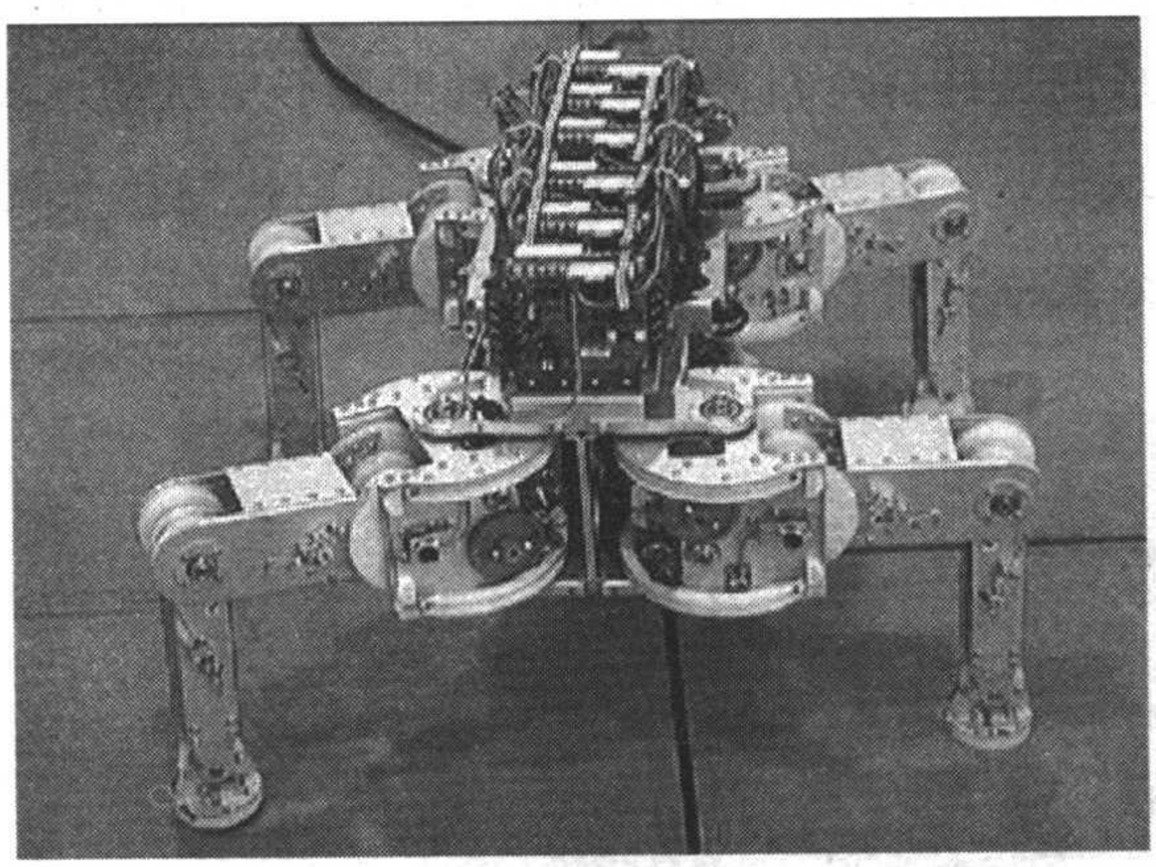


① 立体声话筒：允许 AIBO 拾取周围声音
② 头部传感器：感知有人在头部拍打 AIBO
③ 模式指示器：指示 AIBO 的操作模式
④ 眼灯：亮成蓝绿或红色，指示 AIBO 的情绪
⑤ 彩色摄像机：通过颜色和动作，允许 AIBO 搜索对象，并用颜色和运动认识他们
⑥ 喇叭：发出各种音调和声音效果
⑦ 下巴传感器：感知有人触摸下巴
⑧ 暂停按钮：按下激活或暂停 AIBO
⑨ 胸灯：给出有关机器人状态的信息
⑩ 脚爪传感器：装在各爪的底部
⑪ 尾灯：亮蓝色或是橙色，指示 AIBO 的情绪状态
⑫ 背部传感器：感知有人触摸 AIBO 的背部

图 2.15 日本索尼公司制造的人造狗 AIBO

球。在需要再充电之前，机器人至多能工作一个小时。该机器人初期销量曾经很旺，第一年就卖了60,000多台。可是，这台机器狗所用的电机的数量和技术投资，使得它的售价高达约1500美元。

四腿机器人在人-机交互研究中，具有当作有效人造物的潜能(图2.16)。比如，人类可以像对待宠物一样对待这个索尼机器人，并可以像人和狗那样发展感情联系。此外，索尼已设计了 AIBO 的行走姿态与模仿学习和熟化的一般行为，以随时产生动态的特征行为，这对追踪行为变化的物主来说更加有意义。在解决了高能存储技术和电机工艺难题后，四足机器人可能比 AIBO 有更强的功能，会在全人类环境中变得普遍。

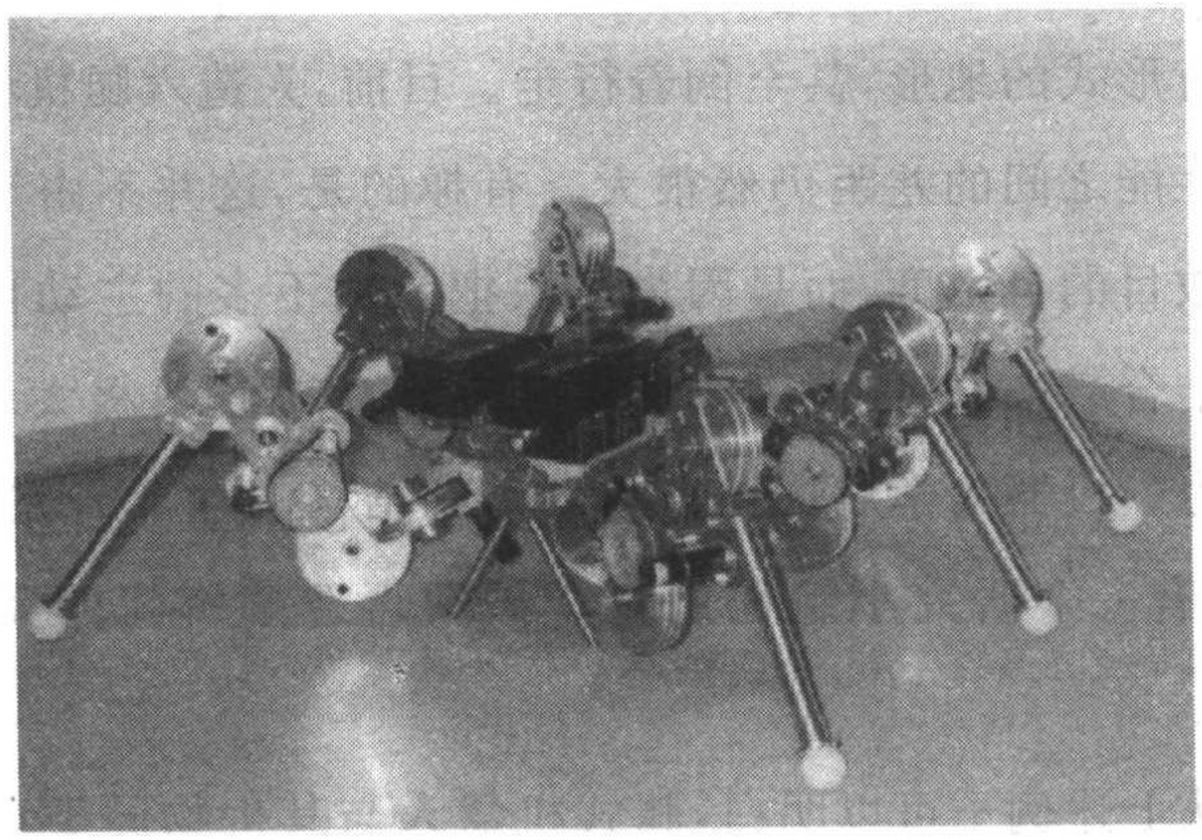


规格：
重量：9 kg
高度：0.25 m
自由度：4×3

图 2.16 东京理工大学开发的四腿机器人 Titan Ⅷ。(http://mozu.mes.titech.ac.sp/research/walk/) © Tokyo Institute of Technology

2.2.2.4 六腿机器人(六脚)

由于行走期间的静态稳定性降低了控制复杂性，六腿结构在移动机器人学中曾经很流行(图 2.17 和图 1.3)。在大多情况下，各条



规格：
最大速度：0.5 m/s
重量：6 kg
高度：0.3 m
长度：0.7 m
腿的数目：6
自由度总数：6×3
功耗：10 W

图 2.17 德国 Karlsruhe 大学开发的六腿平台 Lauron Ⅱ。© University of karlsruhe

腿有 3 个自由度，包括臀部弯曲、膝盖弯曲和臀部外展(图 2.6)。Genghis 是一个商业上已可用的有 6 条腿的业余机器人，各腿有业

余伺服电机所提供的2个自由度(图2.18)。仅有臀部弯曲和臀部外展组成的这样一个机器人,在粗糙的地形中机动性较差,但在平地上则表现很好。因为它是由直腿和伺服电机简单装配而成,所以机器人业余爱好者可以很容易地制作这种机器人。

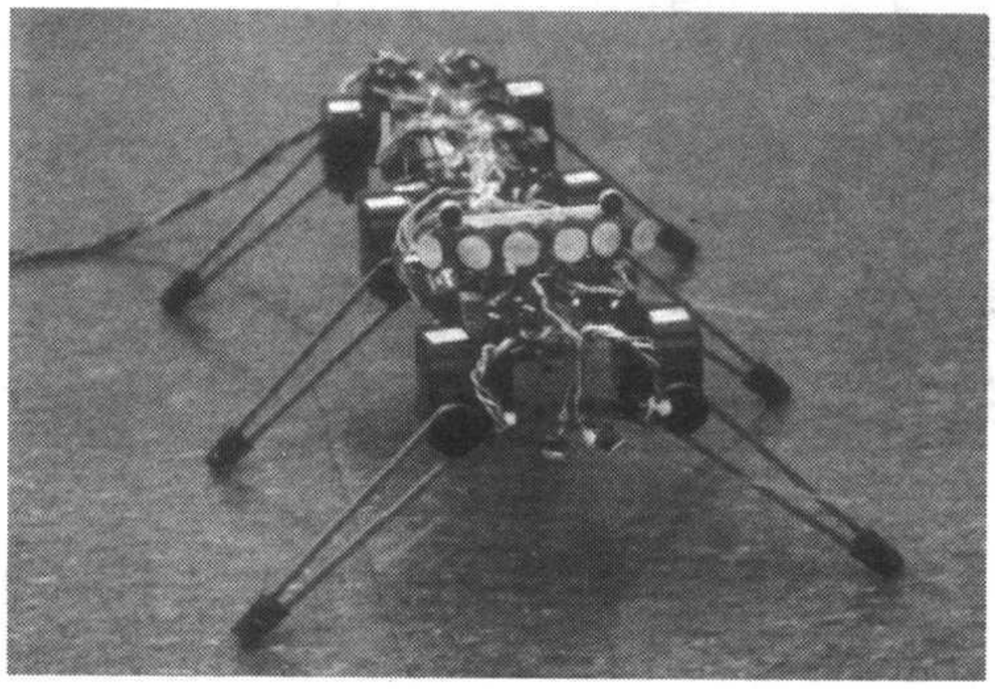


图2.18 Genghis是MIT制作的最有名的行走机器人之一,它用业余伺服电机作激励器(http://www.ai.mit.edu/projects/genghis)。© MIT AI Lab

昆虫被证明是地球上最成功的运动生物。它们擅长于用六足穿越所有形式的地形,甚至倒着行走。目前,人造六腿机器人与六腿昆虫的性能之间的差距仍然很大。有趣的是,这并不由于机器人缺乏足够数目的自由度,而是因为昆虫把为数不多的主动自由度与被动结构结合起来,如细微倒毛和质地粗糙的肉趾,这极大地增强了各腿的抓力。机器人学对这种被动的末端结构的深入研究才刚刚开始。例如,有一个研究小组他们正在试图再造蟑螂腿的完整的机械功能[65]。

从上述例子可以明白,腿式机器人与其生物同类竞争之前还有许多地方需要改进。不过,最近已取得了意义重大的成果,这主要是由于电机设计的进展。要创造一个具有动物肌肉效率的激励系统,目前机器人学达到的水平仍相距甚远,在能量存储方面也远不如生物体的能量密度。

2.3 轮式移动机器人

迄今为止，轮子一般是移动机器人学和人造交通车辆中最流行的运动机构。它可达到很高的效率，如图 2.3 所示，而且用比较简单的机械就可实现它的制作。

另外，在轮式机器人设计中，平衡通常不是一个研究问题。因为在所有时间里，轮式机器人一般都被设计成在任何时间里所有轮子均与地接触。因而，3 个轮子就足以保证稳定平衡。虽然我们将在下面看到，两轮机器人也可以稳定。如果使用的轮子多于 3 个，当机器人碰到崎岖不平的地形时，就需要一个悬挂系统以使所有轮子保持与地面接触。

轮式机器人研究倾向于把重点放在牵引、稳定性、机动性及控制问题，而不是担心平衡。为覆盖所有预期的地形，机器人的轮子能否提供足够的牵引力和稳定性？机器人的轮子结构能对机器人的速度进行有级的控制吗？

2.3.1 轮子运动：设计空间

正如将要看到的，当我们考虑移动机器人运动的可能技术时，可能的轮子结构有很大的空间。因为有很多数目不同的轮子类型，各有其特定的优点和缺点，故我们从详细讨论轮子开始，然后来检验为移动机器人传送特定运动形式的完整的轮子构造。

2.3.1.1 轮子的设计

有四种主要的轮子类型，如图 2.19 所示。在运动学方面，它们差别很大。因此轮子类型的选择对移动机器人的整个运动学有很大的影响。标准轮和小脚轮有一个旋转主轴，因而是高度有向的。在不同的方向运动，必须首先沿着垂直轴操纵轮子。这两种轮的主要

差别在于标准轮可以完成操纵而无副作用，因为旋转中心经过接触片着地；而小脚轮绕偏心轴旋转，在操纵期间会引起一个力，加到机器人的底盘。

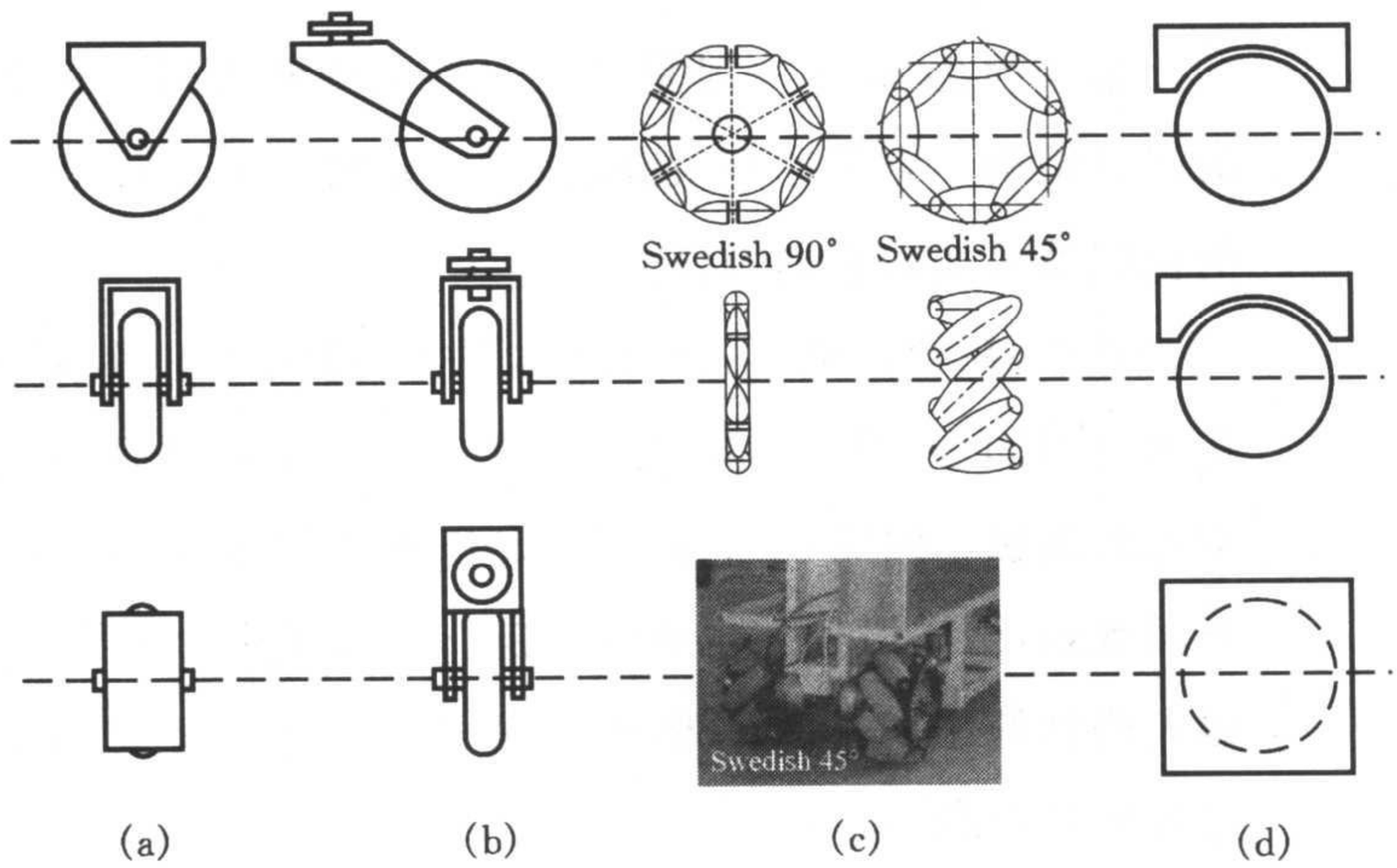


图 2.19 四种基本的轮子类型：(a)标准轮：2个自由度，围绕轮轴(电动的)和接触点转动；(b)小脚轮：2个自由度，围绕偏移的操纵接合点旋转；(c)瑞典轮：3个自由度，围绕轮轴(电动的)、辊子和接触点旋转；(d)球体或球形轮：技术上实现困难

瑞典轮和球形轮二者的设计比传统的标准轮受方向约束少一些。瑞典轮的功能与标准轮一样，但它在另一方向产生低的阻力。该方向有时垂直于常规方向，如 Swedish 90；有时在中间角度，如 Swedish 45。装在轮子周围的辊子是被动的，轮的主轴是唯一的供有动力的连接。这个设计的主要优点在于：虽然轮子旋转仅沿主轴供动力(通过轮轴)，轮子以很小的摩擦，可以沿许多可能的轨迹按运动学原理移动，而不仅仅是向前或者向后。

球形轮是一种真正的全向轮，经常被设计成可以沿任何方向受动力而旋转。实现这种球形构造的一种机构模仿了计算机鼠标，提供了供动力的辊子，这些辊子搁在球的顶部，并给予旋转的力。

无论用什么轮，在为所有地形环境设计的机器人和具有3个以

上轮子的机器人中，正常情况下需要一个悬挂系统以保持轮子与地面的接触。一种最简单的悬挂方法是轮子本身设计成柔性的。例如，在某些使用小脚轮的四轮室内机器人情况下，制造厂家已经把软橡胶可变形的轮胎用在轮上，制作基本的悬挂体。当然，这种有限的解决方案不能与应用中错综复杂的悬挂系统相比拟。在应用中，对明显的非平坦地形，机器人需要更动态的悬挂系统。

2.3.1.2 轮子几何特征

移动机器人轮子类型的选择与轮子装配或轮子几何特征的选择紧密相关。移动机器人的设计者在设计轮式机器人的运动机构时，必须同时考虑这两个问题。为何轮子的类型和轮子的几何特征如此重要？因为机器人的三个基本特征由这些选择所支配：机动性，可控性和稳定性。

汽车大都为高度标准化的环境（公路网络）而设计，与其不同的是，移动机器人则是为应用在种类繁多的环境而设计。汽车全部共享相同的轮子结构，因为在设计空间中存在一个区域，使得它们对标准化环境（铺好的公路）的机动性、可控性和稳定性最大。可是，不同的移动机器人面临各种不同环境，没有单一的轮子结构可以使这些品质最大化。所以，你会看到移动机器人的轮子结构种类繁多。实际上，除了为道路系统设计的移动机器人外，很少机器人使用汽车的Ackerman 轮子结构，因为它的机动性较差（图 2.20）。

表 2.1 给出了轮子结构的概述，按轮子数目排序。表中描述了特殊轮子类型的选择和机器人底盘上它们几何结构的两个方面。我们注意到，所示的某些轮子结构在移动机器人的应用中很少用到。例如，两轮自行车装配，其机动性中等，可控性差。像单腿跳跃机，它根本不能静止地站着。不过，表中提供了在运动机器人设计中可能用到的许多种类轮子结构的说明。

表 2.1 中种类的数目是很多的。不过，这里列出了重要的趋向和分组，它可帮助我们理解各结构的优点和缺点。下面，根据以前确

图 2.20 Navlab Ⅰ是第一个自主式高速公路交通工具。它使用视觉和雷达传感器操作和控制油门[61],由 CMU 开发

认的三个问题:稳定性、机动性和可控性,我们来鉴别一下某些重要的折衷。

2.3.1.3 稳定性

令人惊奇的是,静态稳定所要求的最小轮子数目是 2 个。如上所述,如果质心在轮轴下面,一个两轮差动驱动的机器人可以实现静态稳定。Cye 就是使用这种轮子结构的商业移动机器人(图 2.21)。

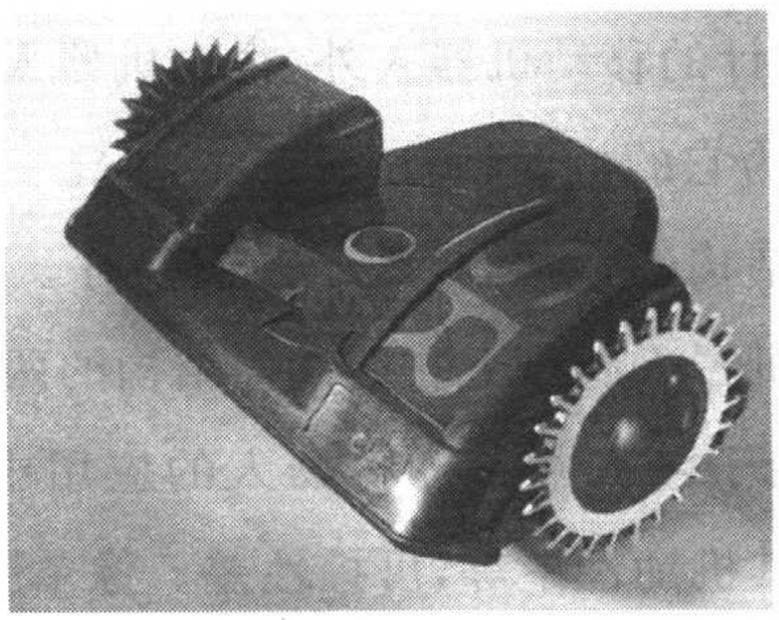


图 2.21 Cye,现有商业化的家用机器人,它可以在家里做真空吸尘和传递员。由 Aethon Inc. (http://www.aethon.com)制造。© Aethon Inc

表 2.1 滚动交通工具的轮子的结构

轮子数目	结构装配	描述	典型例子
2		前端一个操纵轮，后端一个牵引轮	自行车，摩托车
		两轮差动驱动，质心(COM)在转轴下面	Cye 个人机器人
3		带有第 3 个接触点的，两轮居中的差动驱动	Nomad Scout，smart-Rob EPFL
		在后/前端有 2 个独立驱动轮，在前/后端有 1 个全向的无动力轮	许多室内机器人，包括 EPFL 机器人 Pygmalion 和 Alice
		后端有 2 个相连的牵引轮(差动)，前端有 1 个可操纵的自由轮	Piaggio 微型卡车
		后端有 2 个自由轮，前端有 1 个可操纵的牵引轮	Neptune(卡内基梅隆大学)，英雄-1
		3 个动力瑞典轮或球型轮排列成三角形，可以全向运动	斯坦福轮 Tribolo EPFL，掌上导航机器人包(CMU)
		3 个同步的动力和可操纵轮，方向是不可控的	“同步驱动”Denning MRV-2，乔治理工大学，I-Robot B24，Nomad 200
4		后端有 2 个动力轮，前端有两个可操纵轮；两轮操纵必须不同，避免滑动/打滑	后轮驱动的小车

续表 2.1

		前端有 2 个可操纵的动力轮，后端有 2 个自由轮；两轮操纵必须不同，避免滑动/打滑	前轮驱动的小车
		4 个可操纵的动力轮	四轮驱动，四轮操纵 Hyperion(CMU)
		后/前端 2 个牵引轮（差动），前/后端 2 个全向轮	Charlie（DMT - EPFL）
		4 个全向轮	卡内基梅隆大学 天王星
		具有附加接触点的两轮差动驱动	EPFL Khepera，Hyperbot Chip
		4 个动力和可操纵的小脚轮	Nomad XR4000
6		排列在中央的 2 个动力和可操纵轮，四角各有一个全向轮	第一个首次出现
		中央有 2 个牵引轮（差动），四角各有一个全向轮	Terregator（卡内基梅隆大学）
各种轮子类型的图标如下：			
	非动力全向轮（球形，回旋，瑞典）		
	动力瑞典轮（斯坦福轮）		
	非动力标准轮		

续表 2.1

	动力标准轮
	可操纵的动力小脚轮
	可操纵的标准轮
	连接轮

可是，在普通的环境下，这种解决方案要求轮子的直径大得不切实际。动力学也可引起两轮机器人以接触的第 3 个点撞击地面，例如，从静止开始要有足够大的电机转矩。常规情况下，静态稳定要求至少有 3 个轮子，且需要警告的是：重心必须被包含在由轮子地面接触点构成的三角形内。增加更多的轮子可以进一步改善稳定性，虽然一旦接触点超过 3 个后，几何学的超静态性质会要求在有崎岖不平的地形时具有某种形式的灵活的悬挂系统。

2.3.1.4 机动性

一些机器人是全向的，这意味着它们可以在任何时候沿着地平面(x,y)向任意方向运动，包括机器人绕它自己的垂直轴转。这一层次的机动性需要能朝一个以上方向运动的轮子。所以，全向机器人经常使用有动力的瑞典轮或球形轮。天王星(Uranus)是个很好的例子，如图 2.24 所示。这个机器人使用 4 个瑞典轮，能独立地旋转和平移且不受限制。

一般来说，带有瑞典轮和球形轮的机器人由于构造全向轮的机械上约束，其地面清洁度有某些限制。在解决这种地面清洁度问题的同时，一个令人感兴趣的最新解决全向导航的方案是四小脚轮结构。在这种结构中，各小脚轮主动地被操纵，且平移机器人是真正全向的。因为，即使小脚轮的朝向垂直于行走所期望的方向，通过操纵这些轮子，机器人仍能向期望的方向移动。因为垂直轴偏离了地面接触路径，这个操纵运动的结果就是机器人的运动。

在研究领域中，可实现高度机动性的其他类型的移动机器人是很普遍的，它们只比全向结构的机器人稍微差一点。这种类型的机器人，在特定方向上的运动可能开始需要一个旋转运动。机器人的中央有一个圆底盘和转动轴，使得这种机器人可以旋转而不改变它的地面脚印。这种机器人中最普遍的是两轮差动驱动的机器人，在那里2个轮子围绕机器人的中心点转动。为了稳定，根据应用的特点，也许要用1～2个附加的地面接触点。

与上面的结构相反的是我们在汽车中常见的Ackerman操纵结构。这种车辆典型的地方是有一个比小汽车大的旋转直径。而且，它在靠边运动时需要一个由重复向前和向后改变方向组成的停车调动。尽管如此，Ackerman操纵几何结构在业余爱好的机器人市场中仍一直特别地流行。这里，开始用一个遥控的赛车工具包，然后把感知和自治功能加到现有的机械机构，就可以制作机器人了。另外，Ackerman操纵的有限机动性有一个重要的优点：它的定向性和操纵的几何结构，向它提供了在高速旋转中非常好的横向稳定性。

2.3.1.5 可控性

一般来说，可控性和机动性之间存在逆相关性。例如，对全向装置，诸如四小脚轮结构，要求效果显著的处理能把期望的转动和平移的速度转换成单个轮子的命令。而且，这种全向装置经常在轮子上有较大的自由度。例如，瑞典轮沿着轮周有一组自由的辊子。这些自由度造成滑动的积累，导致航位推测准确度的降低并增加了设计的复杂性。

对一个特定的行走方向，控制全向机器人也比较困难，而且当它与较小机动性装置比较时，往往准确度较低。例如，一个Ackerman操纵车辆，通过锁住可操纵轮和驱动它的驱动轮就可以简单地走直线。在差动驱动的车辆情况下，必须精确地按同样的速度分布图驱动装在轮上的2个电机。考虑到轮子间和电机间的差异以及环境的差异，这可能是困难的。对于具有四轮的全向驱动，如Uranus机器

人，它有 4 个瑞典轮，问题更为困难。因为对在理想直线上行走的机器人，必须精确地按相同速度驱动所有的 4 个轮子。

总之，没有“理想”的驱动结构可以同时使稳定性、机动性和可控性最大化。各移动机器人的应用对机器人设计问题加上唯一的约束，而设计者的任务就是在这个折衷空间中，选择最合适的可行的驱动结构。

2.3.2 轮子运动：实例研究

为了展示上面讨论的概念是如何具体应用于为现实环境活动所制造的机器人中，下面我们描述四个特定的轮式结构。

2.3.2.1 同步驱动

在室内移动机器人应用中，同步驱动结构（图 2.22）是一种流行的轮子装配，也是一个令人很感兴趣的结构。因为它虽然有 3 个驱动和操纵轮，可是总共只用了 2 个电机。一个平移电机使 3 个轮的

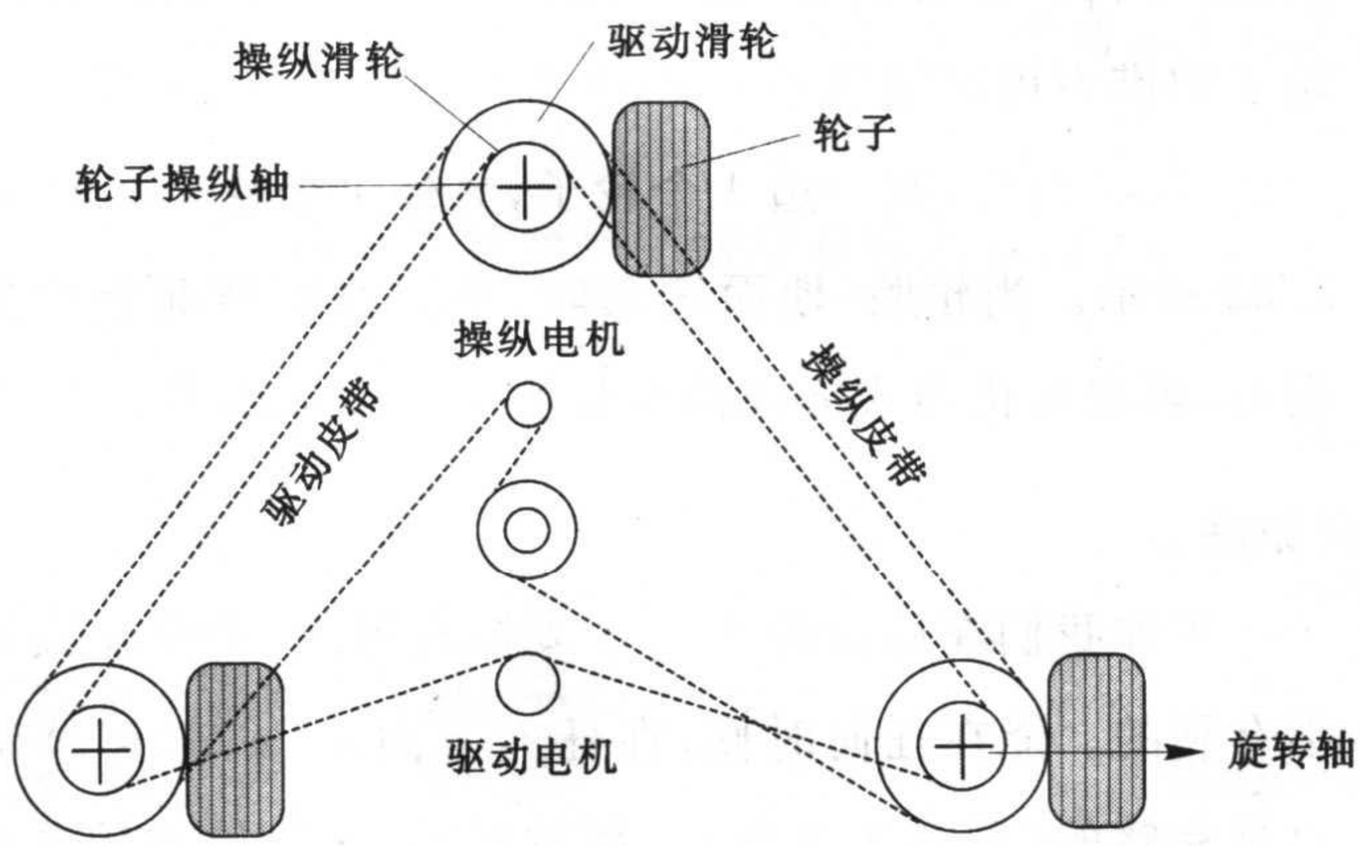


图 2.22 同步驱动：机器人可在任意方向上移动，但是底盘的方向是不可控的

速度一致；一个操纵电机，使所有轮子绕着它们各自的垂直操纵轴一起旋转。但要注意的是，轮子是相对于机器人的底盘受操纵的，因此没有机器人底盘重新定向的直接方法。实际上，由于凹凸不平轮胎

的滑动，底盘方向总是会有一定的漂移，造成旋转航位测定误差。

在追求全向性的情况下，同步驱动特别有好处。只要各垂直的操纵轴与各轮胎的接触路径排列一致，机器人就可以经常对其轮子重新定向，并沿着新轨迹运动而不改变它的脚印。当然，如果机器人的底盘有定向功能，并且设计者有意地打算重新定向底盘，那么当它与一个装在轮子底盘上的独立旋转的转盘结合在一起时，同步驱动则是唯一合适的。商业上的研究机器人如 Nomadics150 或 RWI B21r 都以这种结构出售(图 1.12)。

根据航位测定，同步驱动系统一般优于真正的全向结构，但劣于差动驱动和 Ackerman 操纵系统。对此有两个主要的原因：首先，平移电机一般用单根传送带驱动 3 个轮子。因为驱动链中的泥浆和反冲存在，所以不论何时，当驱动电机起动时，最近的轮子在最远的轮子之前开始旋转，从而引起底盘方向小小的改变。连同附加的电机速度的改变，这些小角度偏移会积累，在航位测定期间产生大的方向误差。其次，移动机器人底盘的方向无直接控制。根据底盘的方向，轮子的推力可以是极不对称的，2 个轮子在一边，第 3 个轮子在单独一边；如果对称，则一边 1 个轮子，另外 1 个轮子在前头或后面，如图 2.22 所示。当轮胎-地面滑动时，不对称的情况会产生各种类型的误差，再次在机器人方向的航位测定中造成误差。

2.3.2.2 全向驱动

正如我们在后面的 3.4.2 节将看到的，全向运动对全机动性是很有利的。能在任何时候，在任何方向上(x,y,θ)移动的全向机器也是完整的(见 3.4.2 节)。用球形轮、小脚轮或瑞典轮可以实现全向机器人。下面给出这种完整的机器人的三个例子。

3 个球形轮的全向运动 图 2.23 所描述的全向机器人，是基于 3 个球形轮子的，它们各由 1 个电机激励。在这种设计中，3 个接触点将球形轮悬挂起来，其中 2 个点由球形轴承给定，另 1 个由连接到电机轴的轮子给定。这种概念提供了极好的机动性，且设计简单。

然而它只限于平坦的路面和小路，制造摩擦系数大的圆形轮子是十分困难的。

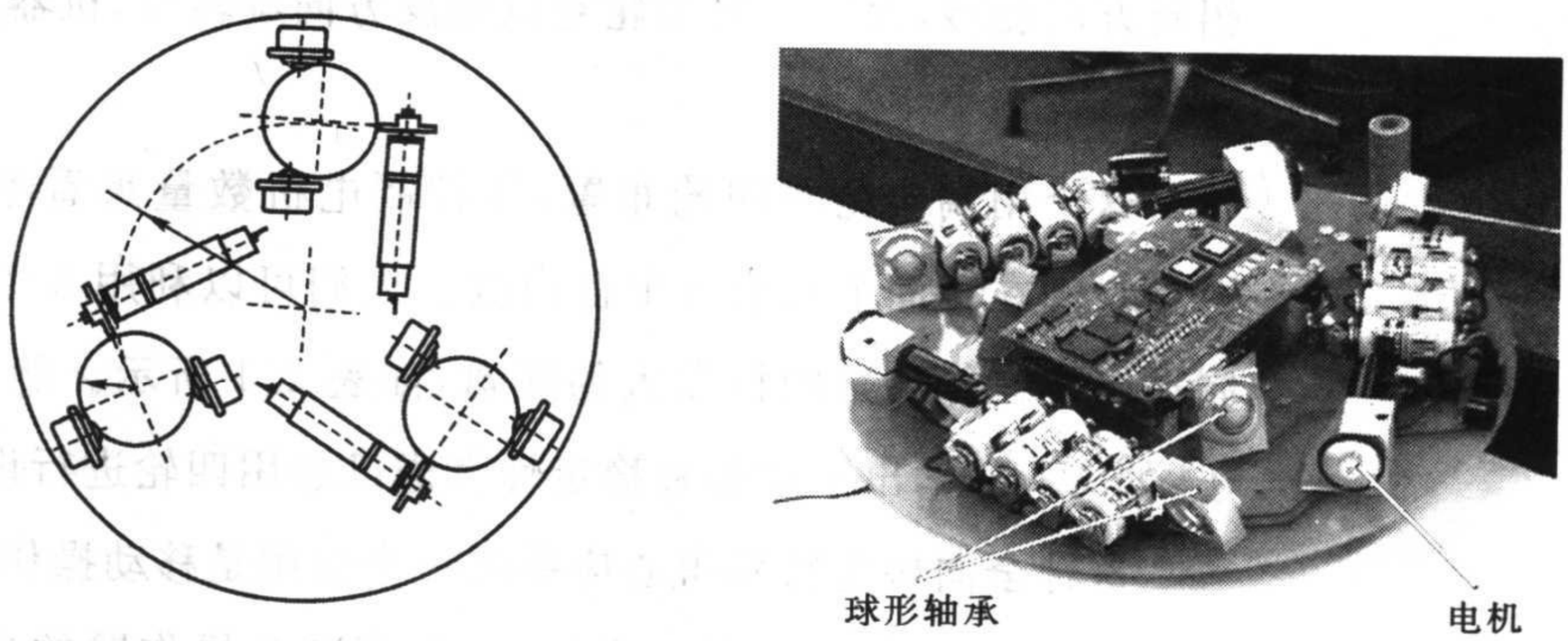


图 2.23 瑞士联邦理工大学(EPFL)设计的 Tribolo。左图：球形轴承和电机的装配(底视图)；右图：无球形轮的机器人图片(底视图)

4 个瑞典轮的全向运动 图 2.24 描述的全向装配已成功地用于几个研究的机器人，包括卡内基梅隆的 Uranus。这种结构包括 4 个 45°的瑞典轮，各由单独的电机驱动。改变 4 个轮子的转动的方向和相对速度，机器人可以沿着平面上的任意轨迹移动。更令人印象深刻的是，机器人甚至可同时绕着它的垂直轴转动。

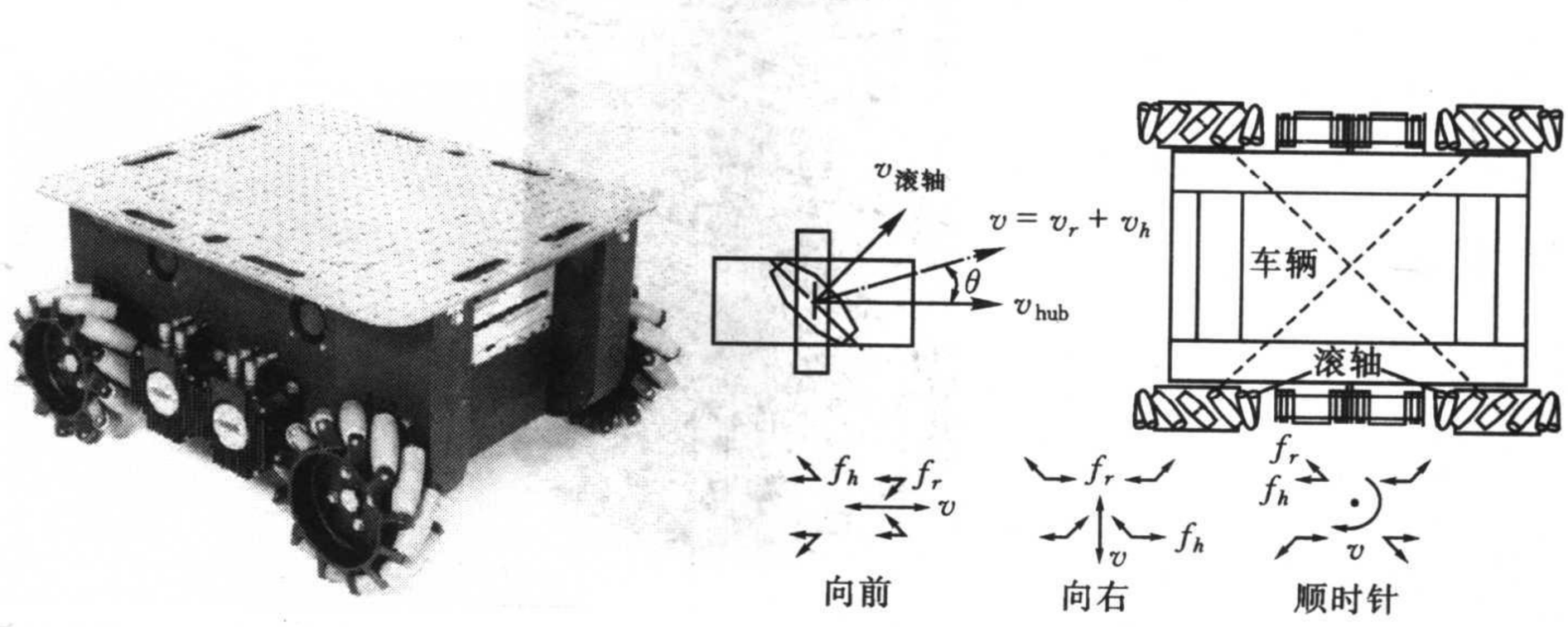


图 2.24 卡内基梅隆的 Uranus 机器人，一个具有 4 个带动力-瑞典 45°轮的全向机器人

例如，当所有4个轮子“向前”或者“向后”旋转时，机器人作为一个整体分别向前或向后在直线上移动。然而，当对角的一对轮子以相同方向旋转，而另一对角轮对以相反方向旋转时，机器人就横向移动。

这个用瑞典轮的四轮布置，从控制电机数量来看不是最少的。因为它们在平面上只有3个自由度。人们可以利用3个90°瑞典轮制造一个三轮的全向机器人的底盘，如表2.1所示。然而现有的例子，如Uranus，出于功能和稳定性考虑已经用四轮进行设计。

这种全向构造特别得心应手的一个应用是移动操作。在这种情况下，利用移动机器人的底盘运动，希望减少操作臂的自由度，且只针对粗略的运动以节省臂的重量。对于人类来说，如果底盘能全向移动，而不很大地影响操纵器顶端的位置，这是理想的。而像Uranus那样的底盘就可精确地提供这种能力。

具有4个小脚轮和8个电机的全向运动 全向性的另一种解决方法是使用小脚轮。Nomadic科技公司（图2.25）的Nomad

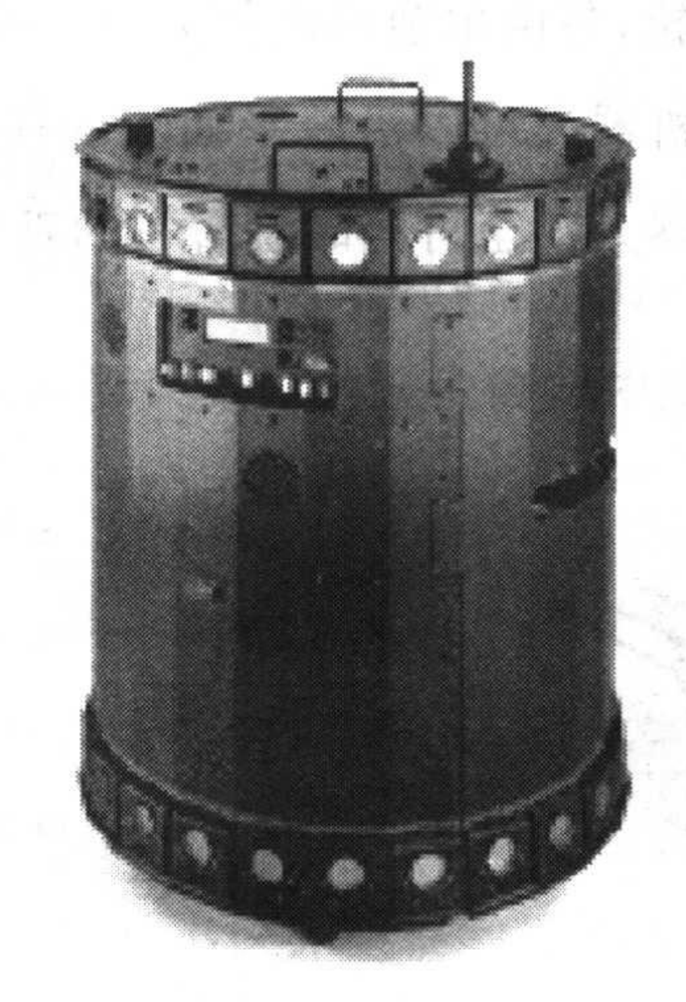


图2.25 Nomadic科技公司的Nomad XR4000有一个针对完整运动的四个小脚轮的装配结构。可驱动和操纵所有小脚轮，因此要求精确的同步和协调，以获得x，y和θ方向的精确运动

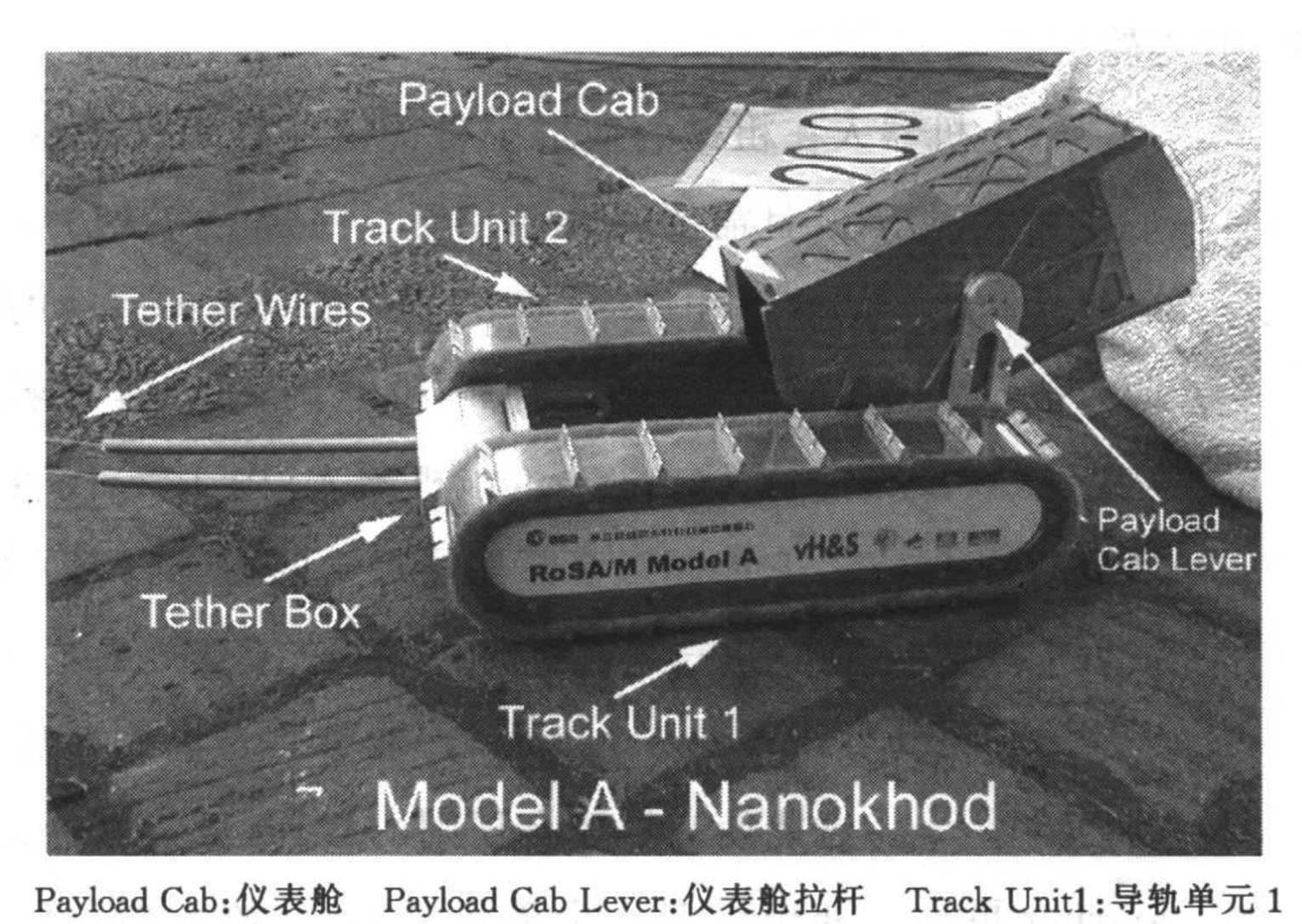
XR4000就是如此，有极好的机动性。遗憾的是，Nomadic已停止生

产移动机器人了。

上面的三个例子取自表 2.1,但该表并不是所有的轮式运动技术的完整列表。腿式和轮式运动相结合或者轨式和轮式运动相结合的混合方法,也可提供特别的优点。下面是为特殊应用而建立的两个独特的构造。

2.3.2.3 轨式滑动/制动运动

在以上讨论的轮子结构中,我们已经作了假定:轮子是不容许对地面打滑。操纵的另一个选择形式叫做滑动/制动,它以不同速度面向相同方向或相反方向旋转轮子,来重新定向机器人。军用坦克用的就是这个方法。Nanokhod(图 2.26)是基于相同概念的移动机器人的一个例子。



Payload Cab:仪表舱 Payload Cab Lever:仪表舱拉杆 Track Unit1:导轨单元 1
Track Unit2:导轨单元 2 Tether Wires:绳索导线 Tether Box:绳索箱
Model A - Nanokhod:模型 A - Nanokhod

图 2.26 微型漫游者 Nanokhod,由美因兹 von Hoerner & Sulger GmbH 和 the Max Planck 研究所为欧洲航天局 (ESA)开发,可能送到火星上[138,154]

使用轮胎面的机器人具有大得多的地面接触面，相比常规的轮子设计，这可以大幅度地改善在松散地形中的机动性。然而，由于这个大的地面接触面，改变了机器人定向时通常要求的滑动旋转，由此，轨迹的大部分必定会相对于地形打滑。

这种结构的缺点与滑动/制动操纵相关。因为旋转中大量打滑，机器人的旋转中心很难准确预测。而且，位置和方向的确切变化也依赖于地面摩擦的变化。所以在这种机器人上，航位测定是很不精确的。回过来，要在极好的机动性和粗糙松软地形上的牵引力之间做出折衷。而且在摩擦力很大的表面上，滑动/制动方法可以很快地超过所用电机的力矩容量。按功效，这种方法在松散地形上是合理有效的；但是在其他情况下是极无效的。

2.3.2.4 行走轮

行走机器人在粗糙的地形中可提供最好的机动性。然而，它在平地上效率差，并且需要复杂的控制。混合的解决方案是将腿的自适应性和轮子的效率相结合，提供了一个令人感兴趣的折衷。被动适应地形的解决方案，对野外的和空间的机器人特别有意义。NASA/JPL 的旅居者(Sojourner)机器人(图 1.2)代表了这样一种混合的解决方式，它能越过高至轮子尺寸的物体。为了相似的应用，最近 EPFL 已生产了一个更新的移动机器人(图 2.27)。这个机器人叫 Shrimp(小虾)，有 6 个动力轮，能够爬过高到其轮子直径 2 倍的物体[97,133]。这能使它攀登有规则的楼梯，虽然机器人甚至比 Sojourner 更小。Shrimp 是一个菱形结构，它的前端和后端各有 1 个操纵轮，2 个轮子安装在各侧的转架上，前轮有个弹簧悬架，保证在任何时刻所有轮子都有最好的地面接触。对漫游者的操纵是通过将前后轮子的操纵同步和转架轮的速度差来实现的。它考虑了高精度的机动性，并以 4 个中心轮最小的滑动/制动实现就地转动。它还使用前轮和转架的并行连接，在轮轴水平面产生一个虚拟的旋转中心。即使轮和地面间摩擦系数很低，也能保证最大的稳定性和爬行能力。

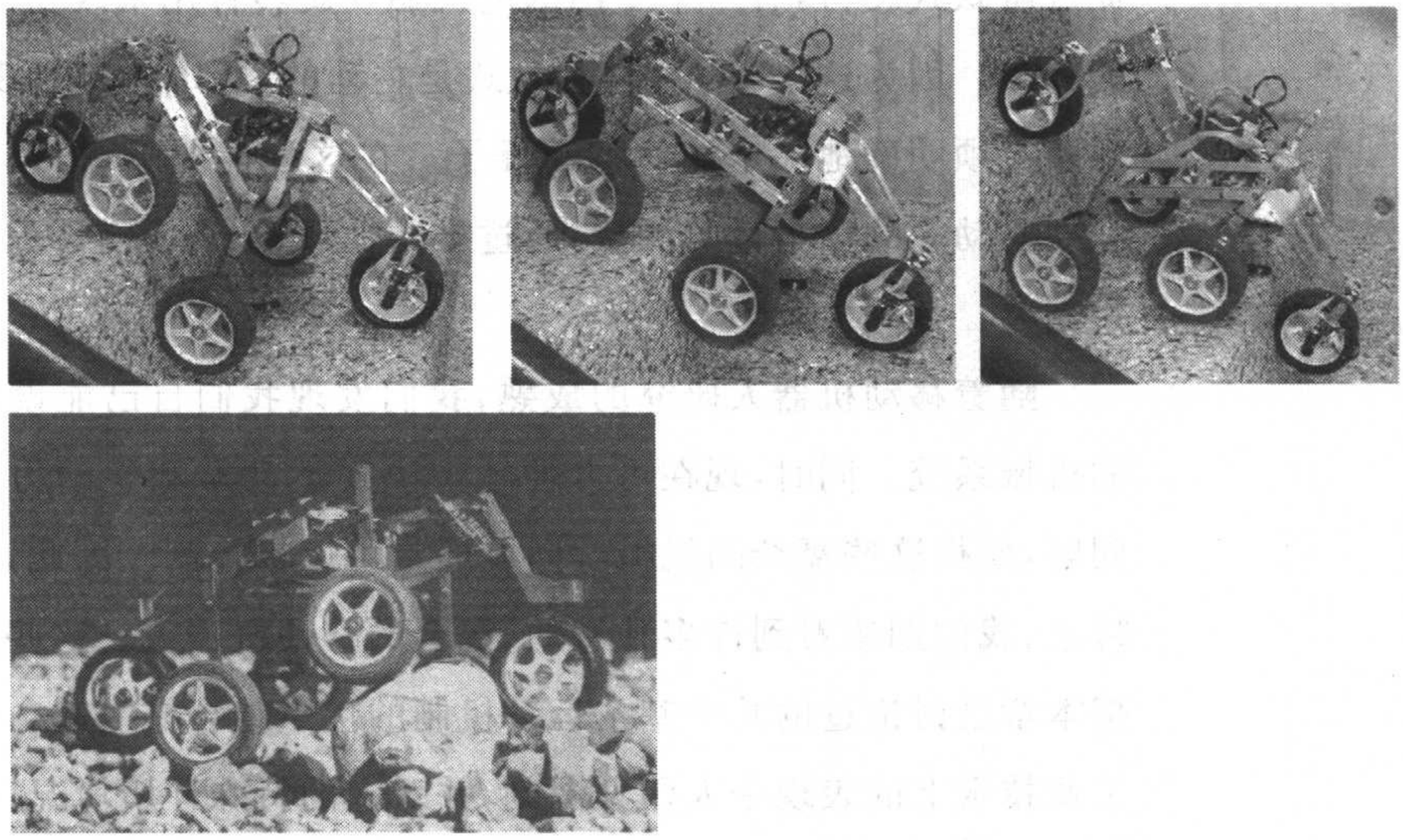


图 2.27 Shrimp，具有卓越的被动爬行能力的全地形机器人（EPFL[97,133]）

与大多数相似机械复杂性的机器人相比，Shrimp 的爬行能力是不同凡响的，这多半由于其特殊的几何特征以及由此产生的机器人质心（COM）可随时相对于轮子移动的方式。相反，个人漫游者展示了攀登台阶的主动的 COM 移动，该台阶也几倍于它轮子的直径，如图2.28所示。旋转杆的上端承受了个人漫游者的大部分重量。为了

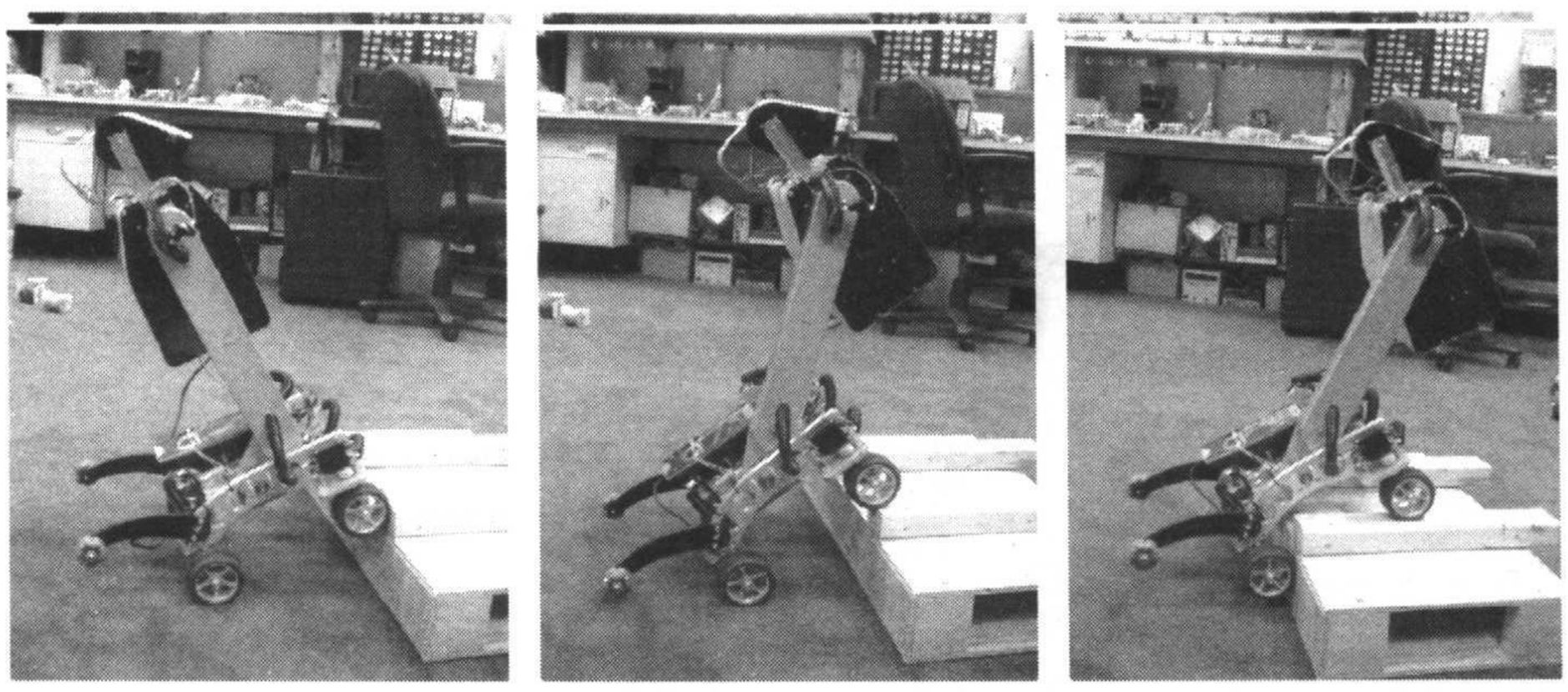


图 2.28 个人漫游者，展示用主动的质心移动爬台阶

促进阶跃式爬行，有一个专门的电机驱动旋转杆用来改变前/后重量的分布。因为这个 COM 移动模式是主动的，所以在爬行现场，控制环路必须明确地决定如何移动杆子。在这种情况下，根据当前到各独立驱动轮的行走量的测量，通过推断地形，个人漫游者完成该闭环控制[66]。

随着移动机器人研究的成熟，我们发现我们自己能设计更复杂的机械系统。同时，现在可方便地解决逆运动学和逆动力学的控制问题，使得这些复杂的机械学问题通常可以被控制。因此，在不远的将来，我们期望看到许多独特的、混合的移动机器人。这些机器人能把本章已讨论过的几个基本运动机制的优点结合在一起。它们各在工业技术上的表现令人印象深刻，并对它的特殊环境境况，各会被设计成专家机器人。

第 3 章　移动机器人运动学

3.1　引言

运动学是对机械系统如何运行的最基本的研究。在移动机器人学中，我们必须了解机器人的机械行为以正确地设计特定任务的移动机器人，并针对一个移动机器人的硬件实例，学会如何创建控制软件。

当然，移动机器人并非是第一个要求这种分析的复杂的机械系统。30 多年来，机器人的机械手一直是集中研究的主题。在某些方面，机器人机械手比早期的移动机器人要复杂得多：一个标准的焊接机械手也许有 5 个甚至更多的关节，而早期的移动机器人只不过是差动驱动的机器。近些年来，机器人学界对机器人机械手的运动学，甚至是动力学（即关于力和质量）已经了解得相当全面了[11,13]。

如同机器人机械手的学界一样，移动机器人学界提出许多相同的运动学难题。机械手的工作空间是至关重要的，因为它定义了相对于它环境的固定装置，以及它的终端执行器可以达到的可能位置的范围。移动机器人的工作空间同样是重要的，因为它定义了在移动机器人的环境中，它能实现的可能姿态的范围。机器人手臂的能控性定义了一种方式，按该方式，可以利用电机的主动接合，在工作空间中一个姿态接一个姿态地运动。相似地，移动机器人的能控性

定义了在它的工作空间中可能的路径和轨迹。由于要考虑质量和力，机器人的动力学在工作空间和轨迹上加上了附加约束。移动机器人也受动力学限制，例如，拟汽车的机器人有翻滚危险高的重心，它限制了高速时的实际转弯半径。

但是移动机器人和机械手臂的重要差别，也引入了一个**位置估计**的重大挑战。机械手有一个固定于环境的终端，测量手臂的位置只不过是为了了解机器人的运动学和测量所有中间关节的位置而已。因此，机械手的位置，通过查看当前传感器数据，常常是能计算的。但是移动机器人是一个独立自动化系统。它能相对于它的环境整体地移动，没有一个直接的方法可以瞬时地测量移动机器人的位置，而是必须随时将机器人的运动集成。此外，由于滑动，**运动估计**不准确，精确地测量移动机器人的位置显然是一个极具挑战性的任务。

了解机器人的运动过程，描述各轮对运动所作的贡献是过程的开始。在整个机器人的运动中，各轮都有作用。相似地，各轮也在机器人的运动中加上约束，例如拒绝横向刹车。在以下章节中，我们将介绍可用全局参考框架和机器人局部参考框架表达机器人运动的概念。然后用该概念，我们展示简单的前向运动的运动学模型，描述机器人作为一个整体，它的几何特征和单个轮子行为的函数是怎样的。接下来，我们从形式上描述单个轮子的运动学约束，然后把这些约束联合起来表述整个机器人的运动学约束。用这些工具，人们可以计算机器人的路径和轨迹，它们定义了机器人的机动性。

3.2 运动学模型和约束

为整个机器人运动推导一个模型，是一个由底向上的过程。各单个轮子对机器人的运动作贡献，同时又对机器人运动施加约束。根据机器人底盘的几何特性，多个轮子是连在一起的。所以它们的

约束联合起来，形成对机器人底盘整个运动的约束。但是，我们必须用相对清晰和一致的参考框架来表达各轮的力和约束。在移动机器人学中，由于它独立和移动的本质，它需要在全局和局部参考框架之间有一个清楚的映射，这一点特别重要。我们从形式上定义这些参考框架开始，然后用最后得到的方程来注释单独轮子和整个机器人的运动学。在这个过程中，我们广泛地引用文献[15]所提出的概念和术语。

3.2.1 表示机器人的位置

在整个分析过程中，我们把机器人建模成轮子上的一个刚体，运行在水平面上。在平面上，该机器人底盘的总的维数是 3 个：2 个为平面中的位置；1 个为沿垂直轴的方向，它与平面正交。当然，由于存在轮轴、轮的操纵关节和小脚轮关节，还会有附加的自由度和灵活性。然而就机器人**底盘**而言，我们只把它看作是机器人的刚体，忽略机器人和它的轮子间内在的关联和自由度。

为了确定机器人在平面中的位置，如图 3.1，我们建立了平面全

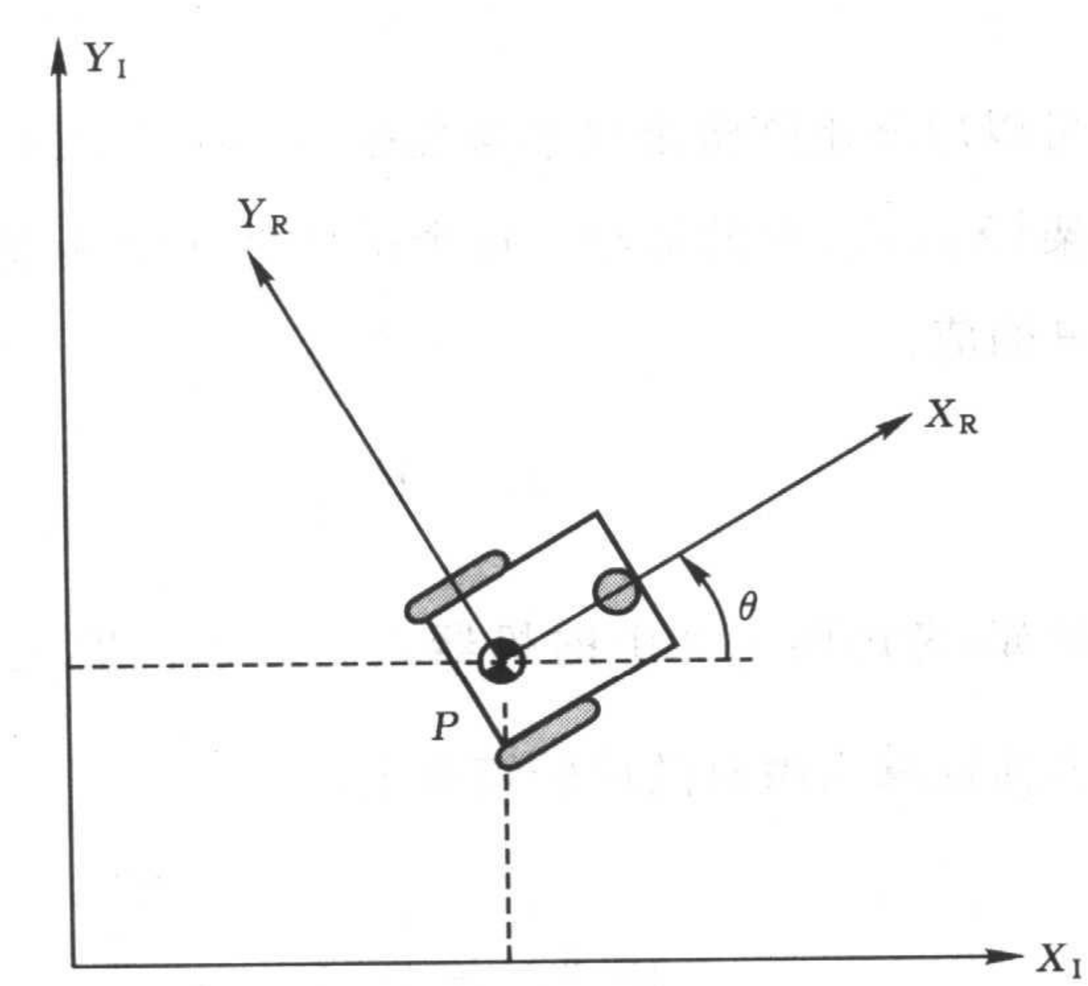


图 3.1 全局参考框架和局部参考框架

局参考框架和机器人局部参考框架之间的关系。轴 X_I 和 Y_I 将平面上任意一个惯性基定义为从某原点 O:$\{X_I, Y_I\}$开始的全局参考框架。为了确定机器人的位置,选择机器人底盘上一个点 P 作为它的位置参考点。基于$\{X_R, Y_R\}$定义了机器人底盘上相对于 P 的两个轴,从而定义了机器人的局部参考框架。在全局参考框架上,P 的位置由坐标 x 和 y 确定,全局和局部参考框架之间的角度差由 θ 给定。我们可以将机器人的姿态描述为具有这 3 个元素的向量。注意,使用下标 I 是为阐明该姿态的基是全局参考框架:

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{3.1}$$

为了根据分量的移动描述机器人的移动,就需要把沿全局参考框架的运动映射成沿机器人局部参考框架轴的运动。当然,映射的是机器人当前位置的函数。该映射用**正交旋转矩阵**来完成:

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

可以用该矩阵将全局参考框架$\{X_I, Y_I\}$中的运动映射到局部参考框架$\{X_R, Y_R\}$中的运动。这个运算由 $R(\theta)\xi_I$ 标记,因为该运算依赖于 θ 的值:

$$\dot{\xi}_R = R(\frac{\pi}{2})\,\dot{\xi}_I \tag{3.3}$$

例如,考虑图 3.2 中的机器人。对该机器人,因为 $\theta = \frac{\pi}{2}$,我们可以容易计算出瞬时的旋转矩阵 R:

$$R(\frac{\pi}{2}) = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

给定在全局参考框架中某个速度$(\dot{x}, \dot{y}, \dot{\theta})$,我们可以计算沿机

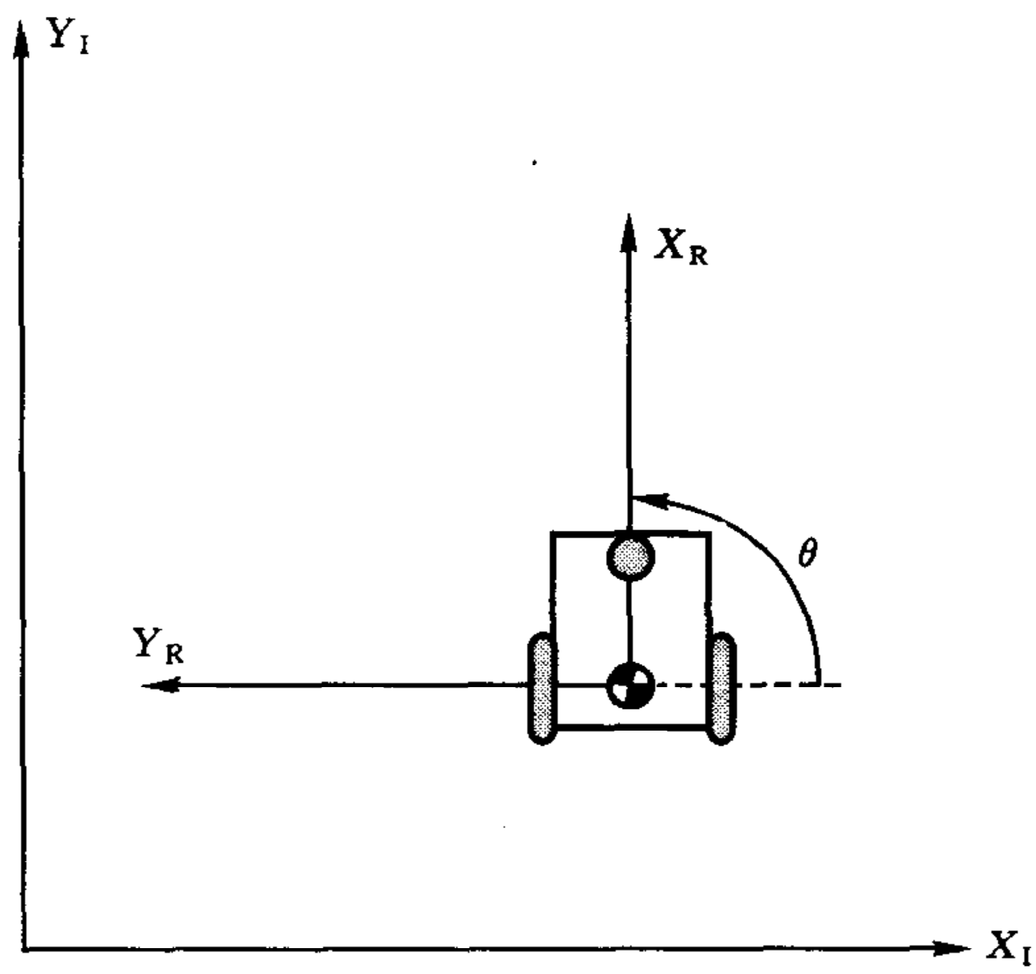


图 3.2 与全局轴并排的机器人

器人局部轴 X_R 和 Y_R 的运动分量。在这情况下，由于机器人的特定角度，沿 X_R 的运动等于$\dot{y}$，沿 Y_R 的运动是$-\dot{x}$：

$$\dot{\xi}_R = R(\frac{\pi}{2})\,\dot{\xi}_I = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{y} \\ -\dot{x} \\ \dot{\theta} \end{bmatrix} \tag{3.5}$$

3.2.2 前向运动学模型

在最简单的情况下，由方程(3.3)所描述的变换足以产生一个获取移动机器人前向运动学的方程：给定机器人的几何特征和它的轮子速度，机器人如何运动呢？更正式地，我们考虑图 3.3 所示的例子。

这个差动驱动的机器人有 2 个轮子，各具直径 r。给定中心处在两轮之间的一个点 P，各轮距 P 的距离为 l。给定 r,l,θ 和各轮的转速$\dot{\varphi}_1$ 和$\dot{\varphi}_2$，前向运动学模型会预测全局参考框架中机器人的总速度：

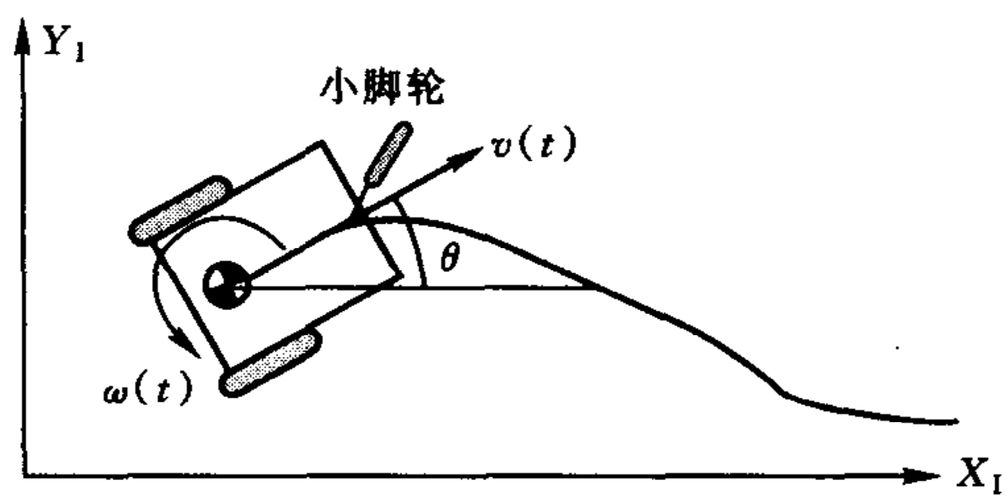


图 3.3 在全局参考框架中差动驱动的机器人

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(l, r, \theta, \dot{\varphi}_1, \dot{\varphi}_2) \tag{3.6}$$

从方程(3.3),我们知道可以从机器人在局部参考框架中的运动计算它在全局参考框架中的运动:$\dot{\xi}_I = R(\theta)^{-1}\dot{\xi}_R$。所以,其策略将是首先计算在局部参考框架中各轮的贡献:$\dot{\xi}_R$。对差动驱动的底盘而言,这个问题是直截了当的。

如图 3.1 所示,假定机器人的局部参考框架,排列得使机器人沿 $+X_R$ 向前运动。首先,我们考虑 $+X_R$ 方向上各轮的转动速度对点 P 的平移速度的贡献。如果一个轮旋转,而另一轮无贡献且是不动的,则因为 P 处在两轮的中间,它将瞬时地以半速移动:$\dot{x}_{r1} = (1/2)r\dot{\varphi}_1$ 和 $\dot{x}_{r2} = (1/2)r\dot{\varphi}_2$。在差动驱动的机器人中,这两个贡献可以简单地相加来计算 $\dot{\xi}_R$ 的 $\dot{x}_R$ 分量。例如,考虑一个差动机器人,其中各轮以等速但反向转动,其结果是一个不动的旋转机器人。如我们所料,在该情况下 $\dot{x}_R$ 将是零。$\dot{y}_R$ 的值计算更为简单。因为在机器人参考框架中,没有一个轮子可以提供侧向运动,所以 $\dot{y}_R$ 总是零。最后,我们必须计算 $\dot{\xi}_R$ 的 $\dot{\theta}_R$ 分量。再次,可以独立地计算各轮的贡献,且只要相加即可。考虑右轮(我们称此为轮 1),该轮向前旋转在点 P 产生逆时针旋转。记得,如果轮 1 单独转动,机器人的枢轴围绕轮 2,可以在点 P 计算旋转速度 ω_1,因为轮子瞬时地沿着半径为 $2l$ 的圆

的圆弧移动：

$$\omega_1 = \frac{r\dot{\varphi}_1}{2l} \tag{3.7}$$

同样的计算施加于左轮，除了向前旋转在点 P 产生顺时针转动之外：

$$\omega_2 = \frac{-r\dot{\varphi}_2}{2l} \tag{3.8}$$

联合这两个方程，得到差动驱动实例机器人的运动学模型：

$$\dot{\xi}_I = R(\theta)^{-1}\begin{bmatrix} \frac{r\dot{\varphi}_1}{2} + \frac{r\dot{\varphi}_2}{2} \\ 0 \\ \frac{r\dot{\varphi}_1}{2l} + \frac{-r\dot{\varphi}_2}{2l} \end{bmatrix} \tag{3.9}$$

现在我们可以在一个例子中使用该运动学模型。但我们必须首先计算$R(\theta)^{-1}$。一般而言，计算矩阵逆可能是难题。然而在这个例子中，因为这是简单的从$\dot{\xi}_R$ 到$\dot{\xi}_I$ 的变换而不是向反，所以是容易的：

$$R(\theta)^{-1} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

假定机器人位于 $\theta=\pi/2, r=1$ 和 $l=1$。如果机器人非均衡地啮合它的轮子，速度$\dot{\varphi}_1=4, \dot{\varphi}_2=2$，我们可以计算它在全局参考框架中的速度：

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix} \tag{3.11}$$

所以这个机器人会沿着全局参考框架的 y 轴，以速度 1 旋转的同时以速度 3 瞬时地移动。在简单的情况下，给定机器人的组成轮的速度，用这个运动学建模的方法可以提供有关机器人移动的信息。然而我们希望，对各机器人底盘结构，要确定可能运动的空间。为

此，我们必须更进一步正式地描述各轮加到机器人运动上的约束。3.2.3 节通过描述各种不同类型轮子的约束，开始了这个过程。在本章的其余部分，为给出这些约束，我们提供了分析机器人特征和工作空间的工具。

3.2.3 轮子运动学约束

机器人运动学模型的第一步是表达加在单独轮子上的约束。正如在 3.2.2 节中所示那样，单独轮子的运动以后可以被联合起来计算整个机器人的运动。如在第 2 章所讨论那样，有四种基本的轮子类型，它们各具变化广泛的运动学参数。所以，我们一开始就要提出对各轮子类型特定的约束集合。

不过，有几个重要的假设会简化上述的陈述。我们假定轮子的平面总是保持垂直，且在所有的情况下，在轮子与地面之间只有一个单独的接触点。此外，我们假定在该单独的接触点无滑动。也就是说，轮子只在纯滚动和通过接触点绕垂直轴转动的条件下进行运动。对运动学更深入的处理，包括滑动接触，请参阅参考文献[25]。

在这些条件下，对每一个轮子类型，我们提出两个约束。第一个约束坚持滚动接触的概念，即当运动在适当方向发生时，轮子必须滚动。第二个约束坚持无横向滑动的概念，即在正交于轮子的平面，轮子必须无滑动。

3.2.3.1 固定的标准轮

固定的标准轮没有可操纵的垂直转动轴。因此，它相对于底盘的角度是固定的，因而限制了沿轮子平面前后运动和围绕与地面接触点的转动。图 3.4 刻画了一个固定标准轮 A，并说明了它相对于机器人局部参考框架$\{X_R, Y_R\}$的位置姿态。A 的位置用极坐标中的距离 l 和角度 α 表示。轮子平面相对于底盘的角度用 β 表示，因为固定标准轮是不可操纵的，所以 β 是固定的。具有半径 r 的轮子可随时转动，所以它围绕它的水平轴转动的位置是时间 t 的函数：$\varphi(t)$。

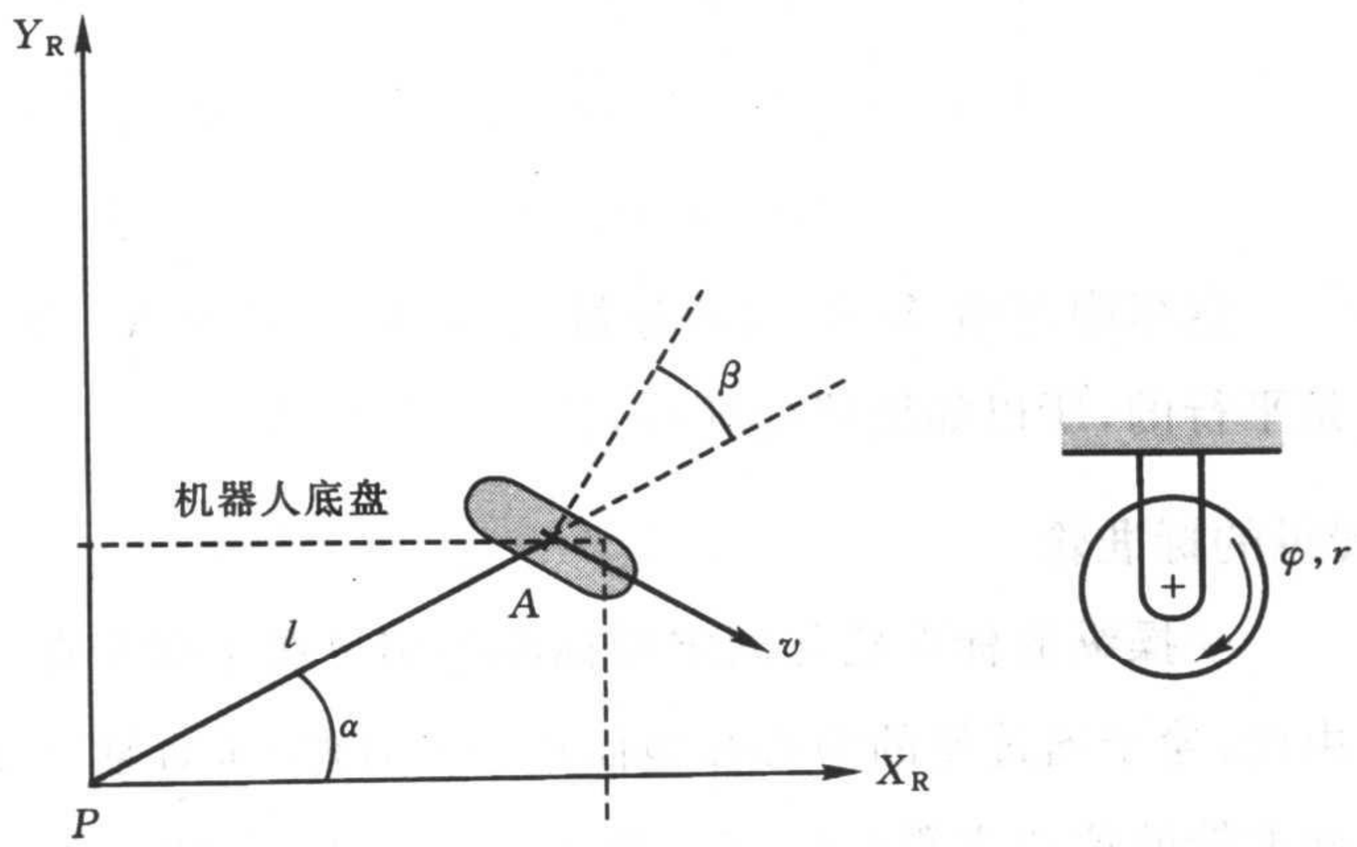


图 3.4 固定标准轮和它的参数

对该轮的滚动约束迫使所有沿轮子平面方向的运动必定伴随着适量的轮子转动，使得在接触点存在纯的滚动：

$$[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad (-l)\cos\beta]R(\theta)\,\dot{\xi}_I - r\dot{\varphi} = 0 \qquad (3.12)$$

和的第一项表示了沿轮子平面总的运动。对于沿着轮子平面的运动，左边向量的三个元素代表了从各个 $\dot{x}$，$\dot{y}$，$\dot{\theta}$ 到它们贡献的变换。注意，如举例方程(3.5)所示，我们用 $R(\theta)\dot{\xi}_I$ 项将处在全局参考框架$\{X_I, Y_I\}$中的参数$\dot{\xi}_I$ 变换到处在局部参考框架$\{X_R, Y_R\}$内的运动参数$\dot{\xi}_R$。这是必须的，因为所有在方程中的其他参数 α，β，l 都是依据机器人的局部参考框架。根据这个约束，沿着轮子平面的运动，必须等于由旋转轮子完成的运动，$r\dot{\varphi}$。

对该轮子的滑动约束，迫使正交于轮子平面的轮子运动分量必须为零：

$$[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta]R(\theta)\,\dot{\xi}_I = 0 \qquad (3.13)$$

例如，假定轮 A 处在一个位置，使$\{\alpha=0, \beta=0\}$，这会把在 X_I 上的接触点，放到面向平行于 Y_I 的轮平面中。如果 $\theta=0$，那么滑动约束(方程(3.13))简化为：

$$[1 \quad 0 \quad 0]\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{x}\\\dot{y}\\\dot{\theta}\end{bmatrix}=[1 \quad 0 \quad 0]\begin{bmatrix}\dot{x}\\\dot{y}\\\dot{\theta}\end{bmatrix}=0 \tag{3.14}$$

这里限定沿 X_I 的运动分量为零，而且因为 X_I 和 X_R 在本例中是平行的，所以如期待的那样，轮子不会滑动。

3.2.3.2 受操纵的标准轮

受操纵的标准轮与固定的标准差别只在于前者有一个附加的自由度：轮子通过轮的中心和地面接触点，围绕垂直轴转动。受操纵的标准轮的位置方程(图3.5)，除了一个例外，其他与图 3.4 所示的固

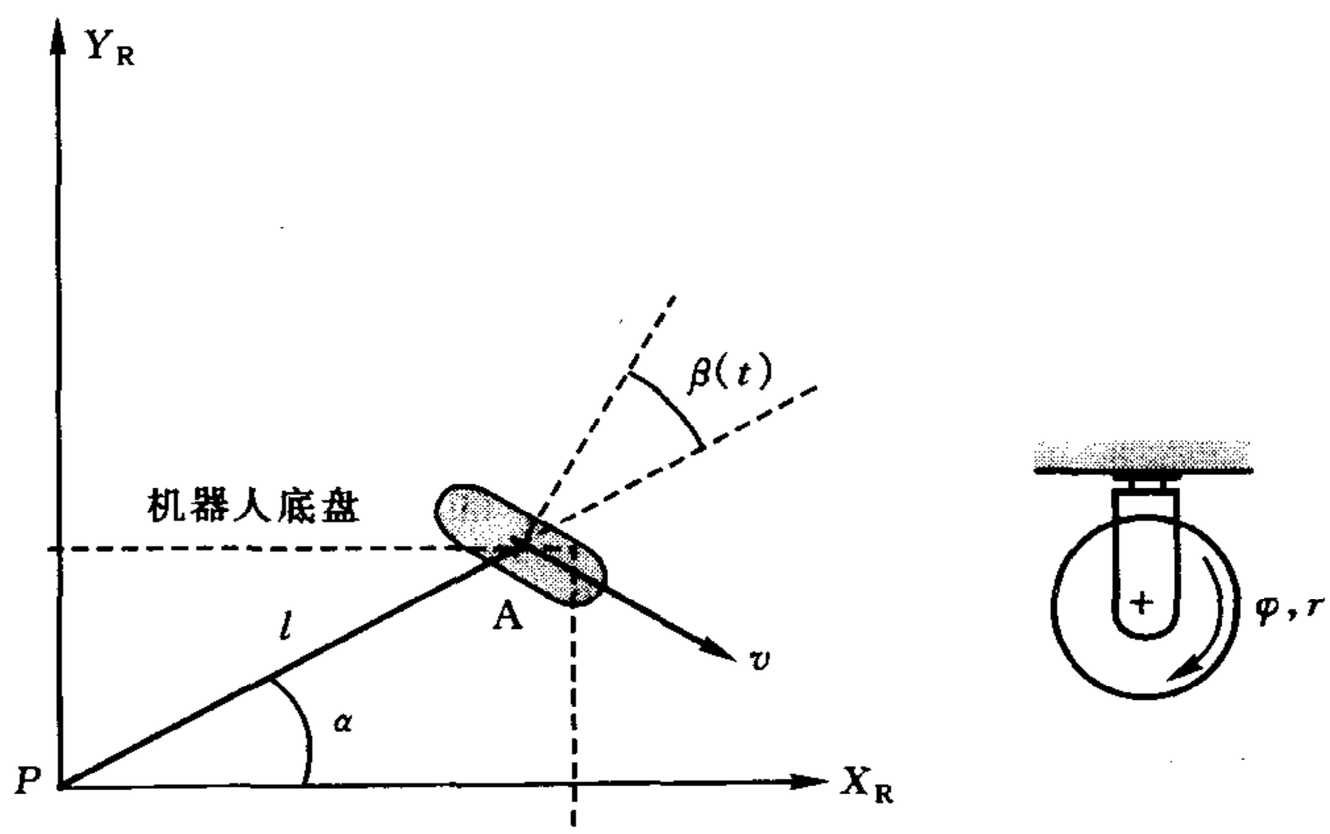


图 3.5 受操纵的标准轮和它的参数

定标准轮是一样的。轮子对机器人底盘的方向不再是一个单独的固定值 β，而是随时间而变的函数：$\beta(t)$。滚动和滑动的约束为：

$$[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad (-l)\cos\beta]R(\theta)\,\dot{\xi}_I - r\dot{\varphi} = 0 \tag{3.15}$$

$$[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta]R(\theta)\,\dot{\xi}_I = 0 \tag{3.16}$$

这些约束与固定标准轮的约束是相同的，不像 $\dot{\varphi}$，$\dot{\beta}$ 对机器人瞬时运动的限制没有直接的影响。仅仅通过随时间的集成，操纵角的变化影响车辆的活动性。这看起来也许是微妙的，但在操纵位置的变化 $\dot{\beta}$ 和轮子旋转 $\dot{\varphi}$ 的变化之间，这是个很重要的区别。

3.2.3.3 小脚轮

小脚轮能绕垂直轴操纵。但与受操纵的标准轮不一样的是，小脚轮绕垂直轴转动不通过地面接触点。图 3.6 表示了一个小脚轮，说明小脚轮位置的正式指标需要一个附加的参数。

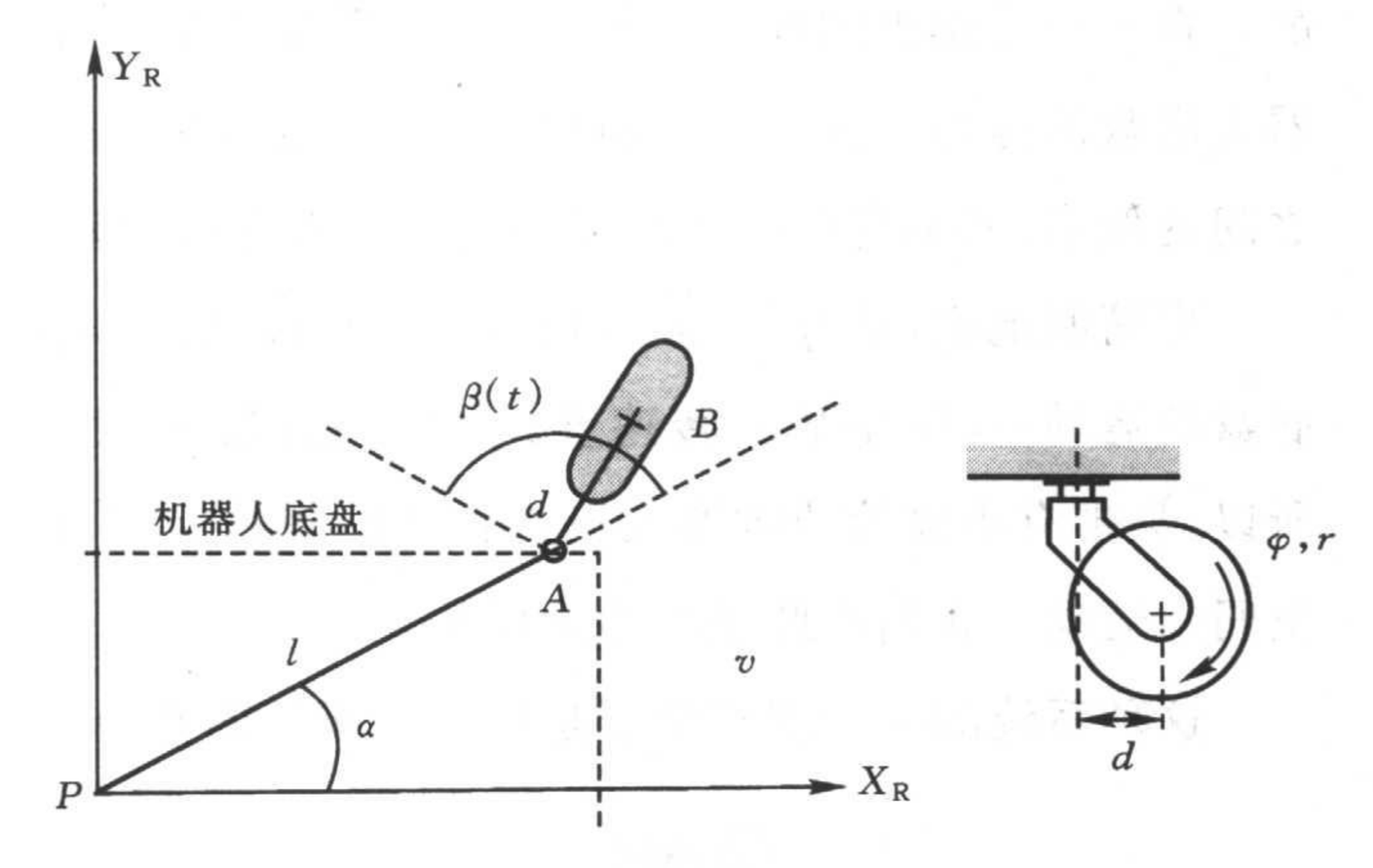


图 3.6 小脚轮和它的参数

现在轮子的接触点在位置 B，它被固定长度 d 的刚性杆 AB 连接到点 A，从而固定了垂直轴的位置，B 围绕该轴进行操纵。如图 3.6 所示，点 A 在机器人的参考框架中有一个特定的位置，我们假定在所有的时间里，轮子平面与 AB 一致。与受操纵的标准轮相似，小脚轮有两个随时间改变的参数，$\varphi(t)$ 如前表示了轮子随时间的转动，$\beta(t)$ 表示了 AB 随时间的操纵角度和方向。

对小脚轮而言，滚动约束与方程(3.15)相同，因为在与轮子平面一致的运动期间，轴偏移不起作用：

$$[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad (-l)\cos\beta]R(\theta)\,\dot{\xi}_I - r\dot{\varphi} = 0 \quad (3.17)$$

然而，小脚轮的几何特征，确实对滑动约束有重大影响。关键性的问题是在点 A 发生对轮子的横向力，因为这是轮子对底盘的接触点。由于相对于 A 点的地面偏移接触点，横向运动为零的约束是错误的，此约束反而很像是一个滚动约束。因此垂直轴的适当转动必定产生：

$$[\cos(\alpha+\beta) \quad \sin(\alpha+\beta)d \quad +l\sin\beta]R(\theta)\,\dot{\xi}_I + d\dot{\beta} = 0 \qquad (3.18)$$

在方程(3.18)中,任何正交于轮子平面的运动必须被一个等效的且相反的小脚轮操纵运动量所平衡。这个结果对小脚轮的成功是至关重要的,因为通过设置$\dot{\beta}$值,任何任意的横向运动是可以被接受的。在一个受操纵的标准轮中,不是由本身产生的操纵动作引起机器人底盘的运动。但是在小脚轮中,由于地面接触点和垂直旋转轴之间的偏移,其本身的操纵动作会使机器人底盘移动。

更简明地说,从方程(3.17)和(3.18)可以猜测,给定**任何**机器人底盘的运动$\dot{\xi}_I$,存在某旋转速度$\dot{\varphi}$和操纵速度$\dot{\beta}$,使得约束被满足。所以,只带有小脚轮的机器人可按任意的速度在可能的机器人运动空间中运动。我们称此系统为**全向的**。

这种系统的一个现实例子是图3.7所示的5个小脚轮的办公室

图3.7 有5个小脚轮的办公椅

椅子。假定所有关节能够自由地运动,你可以在平面上为椅子选择任何运动向量,并且用手推它,它的小脚轮将按需要旋转和操纵,实现运动而无接触点滑动。相似地,如果椅子小脚轮各装上两个电机,一个作旋转;另一个作操纵,那么控制系统就有能力使椅子沿着平面中的轨迹运动。因此,虽然小脚轮的运动学有些复杂,但这种轮子在机器人底盘上,没有加任何实际约束。

3.2.3.4 瑞典轮

瑞典轮没有垂直旋转轴，但能像小脚轮一样地**全向**运动。在固定标准轮上加自由度是可行的。瑞典轮由带滚柱的固定标准轮所组成，滚柱放在带轴的轮子周界上，滚柱的轴逆平行于固定轮组成部分的主轴。滚柱和主轴之间的准确角度是可以变化，如图 3.8 所示。

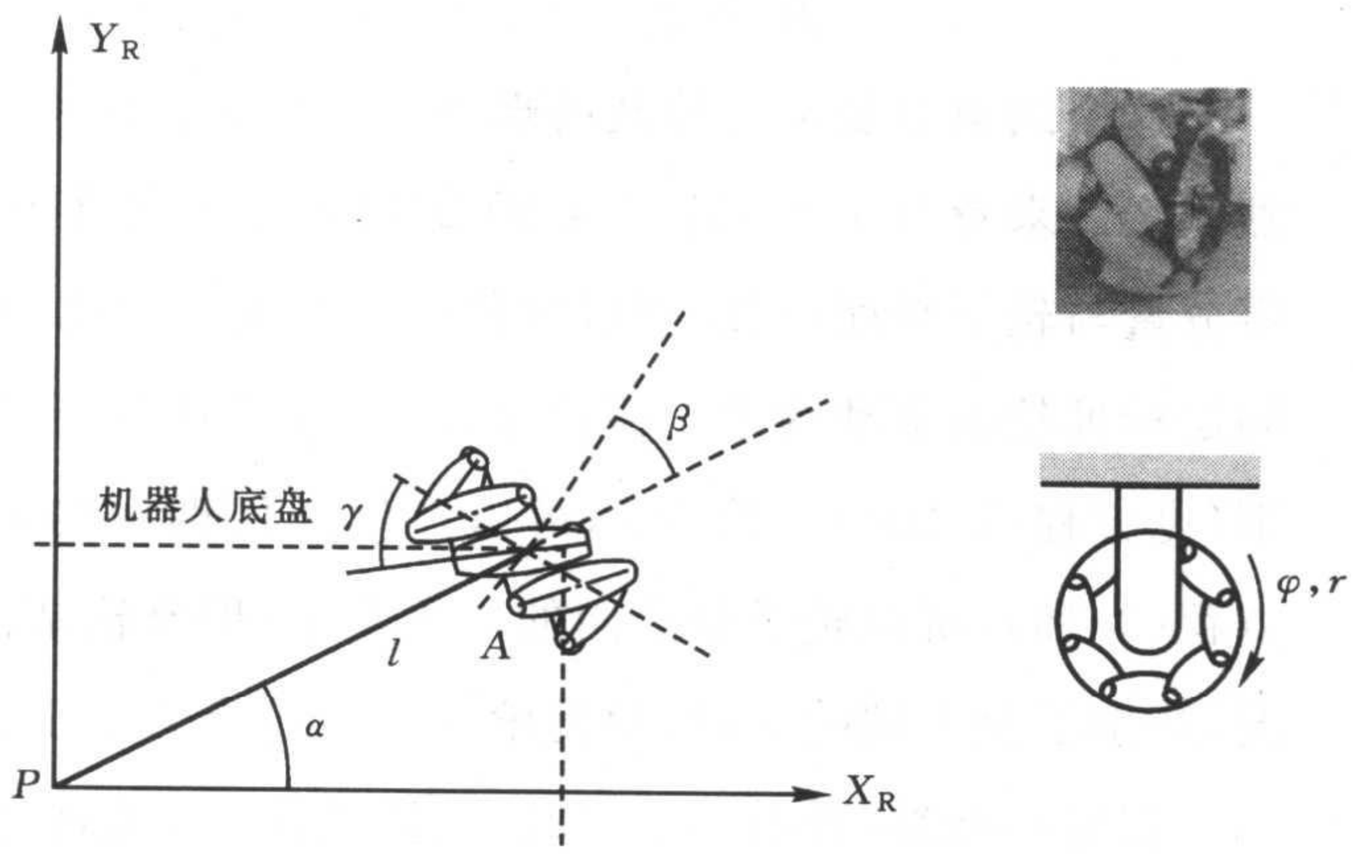


图 3.8 瑞典轮及其参数

例如，给定一个 45°的瑞典轮，如图 3.8，可以画出主轴和滚柱轴的运动向量。因为各轴可以顺时针或逆时针转动，人们可以将沿一个轴的任何向量与沿另一轴的任何向量联合起来。这两个轴不必独立(除了 90°瑞典轮情况外)，但在视觉上看得清楚，任何期望的运动方向，通过选择合适的两个向量是可以实现的。

瑞典轮的姿态准确地被表示成和固定标准轮一样，但带有附加项 γ，它表示主轮平面和小圆周的滚柱旋转轴之间的角度。这表示在图 3.8 中机器人参考框架内。

对瑞典轮建立约束方程需要有某些技巧。瞬时的约束是由于小滚柱的特定方向形成的。围绕这些滚柱，主轴转动是接触点上速度的一个零分量，即在那个特定方向的运动，不转动主轴是不可能没有滑动的。推导运动约束看起来与方程(3.12)中固定标准轮的滚动约束一样，除了加上 γ 修改了方程，使得滚动约束成立的有效方向是沿

着这个零分量，而不是沿着轮子的平面：

$$[\sin(\alpha+\beta+\gamma) \quad -\cos(\alpha+\beta+\gamma) \quad (-l)\cos(\beta+\gamma)] \cdot R(\theta)\,\dot{\xi}_I - r\dot{\varphi}\cos\gamma = 0 \tag{3.19}$$

由于小滚柱的自由转动$\dot{\varphi}_{sw}$，正交于该运动方向不受约束。

$$[\cos(\alpha+\beta+\gamma) \quad \sin(\alpha+\beta+\gamma) \quad l\sin(\beta+\gamma)] \cdot R(\theta)\,\dot{\xi}_I - r\dot{\varphi}\sin\gamma - r_{sw}\,\dot{\varphi}_{sw} = 0 \tag{3.20}$$

这个约束以及由此涉及的瑞典轮的行为特征将随γ值变化而激烈改变。来看看$\gamma=0$，这代表90°的瑞典轮。在这个情况下，速度的零分量与轮子平面一致，所以方程(3.19)正好简化为方程(3.12)，即固定标准轮的滚动约束。但由于滚柱，正交于轮子平面没有滑动约束(见方程(3.20))。改变$\dot{\varphi}$，可以产生任何期望的运动向量以满足方程(3.19)，所以轮子是全向的。事实上，瑞典轮结构的这个特殊情况，产生了全去耦的运动，使得滚柱和主轮提供了正交的运动方向。

在另一极端，看看$\gamma=\pi/2$。在这情况下，滚柱有一个平行于主轮旋转轴的转动轴。有趣的是，如果该值取代方程(3.19)中的γ，结果就是固定标准轮的滑动约束，即方程(3.13)。换句话说，根据运动的横向自由度性质，滚柱没有用处，因为它们简单地与主轮方向一致。然而，在这个情况下，主轮决不需要旋转，所以滚动约束消失。这是瑞典轮的退化形式，所以在本章的其余部分，我们都假定$\gamma\neq\pi/2$。

3.2.3.5 球形轮

最后的一个轮子类型，球或球形轮，不对运动加约束(图3.9)。这种机械结构没有转动主轴，所以不存在合适的滚动和滑动约束。如同小脚轮和瑞典轮，球形轮显然是全向的，对机器人的底盘运动学不加约束。所以方程(3.21)简单地描述了在机器人点A的运动方向v_A中球的滚动速率。

$$[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad (-l)\cos\beta]R(\theta)\,\dot{\xi}_I - r\dot{\varphi} = 0 \tag{3.21}$$

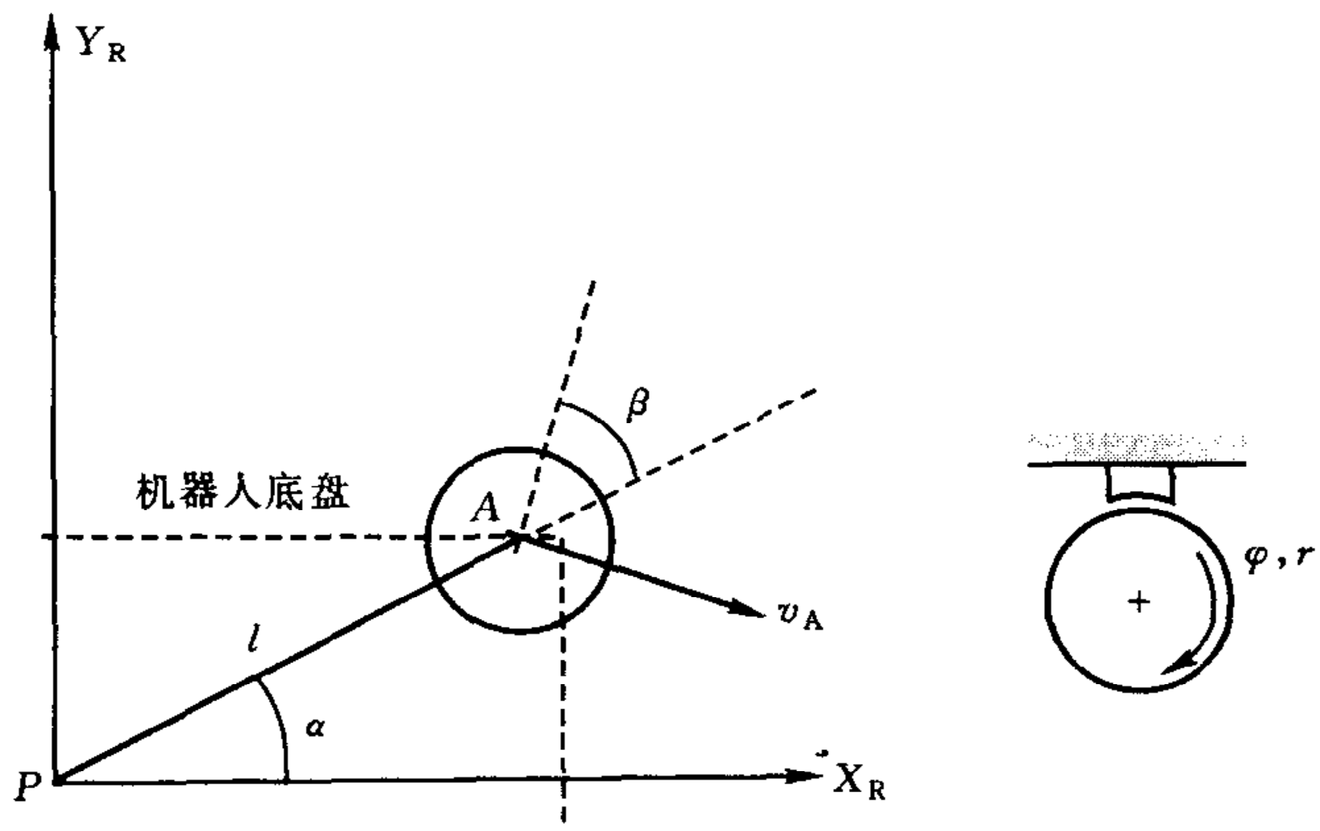


图 3.9 球形轮及其参数

根据定义，正交于该方向的轮子转动是零。

$$[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta]R(\theta)\,\dot{\xi}_I = 0 \qquad (3.22)$$

正如我们所见，球形轮的方程与固定标准轮完全一样。然而，对方程(3.22)的解释是不同的。全向的球形轮可以有任何的运动方向。这里，由 β 给出的运动方向是从方程(3.22)演绎出来的一个自由变量。考虑在 Y_R 方向上机器人是纯平移的情况，这时方程(3.22)简化为 $\sin(\alpha+\beta)=0$。因此，$\alpha=-\beta$，它对该特殊情况有意义。

3.2.4 机器人运动学约束

给定一个具有 M 个轮子的机器人，现在我们可以计算机器人底盘的运动学的约束。主要的思想是各轮子对机器人的运动加上零或更多的约束，所以过程只不过是根据机器人底盘上那些轮子的配置，将全部由所有轮子引起的运动学的约束适当地联合起来。

我们已经把全部的轮子分类成五种类别：(1)固定的标准轮，(2)可操纵的标准轮，(3)小脚轮，(4)瑞典轮，(5)球形轮。但是，从方程(3.17)、(3.18)和(3.19)中的轮子运动学的约束，注意到小脚轮、瑞典轮和球形轮在机器人底盘上**没有**加运动学的约束，因为由于内部的轮子的自由度，$\dot{\xi}_I$ 在所有这些情况中可自由地变化。

所以，只有固定的标准轮和可操纵的标准轮对机器人底盘的运动学有影响。因而，当计算机器人运动学的约束时，需要予以考虑。假定机器人总共有 N 个标准轮，它由 N_f 个固定的标准轮和 N_s 个可操纵的标准轮组成。我们用 $\beta_s(t)$ 表示 N_s 个可操纵标准轮的可变操纵角；相反，如在图 3.4 所示，β_f 当作 N_f 个固定标准轮的方向。在轮子旋转的情况下，固定的和可操纵的轮子二者都有围绕水平轴的转动位置，它随时间函数而变化。我们把固定和可操纵情况分别表示成 $\varphi_f(t)$ 和 $\varphi_s(t)$，并用 $\varphi(t)$ 表示联合二者值的集结矩阵。

$$\varphi(t) = \begin{bmatrix} \varphi_f(t) \\ \varphi_s(t) \end{bmatrix} \tag{3.23}$$

所有轮子的滚动约束现在可以被集合成一个单独表达式：

$$J_1(\beta_s)R(\theta)\,\dot{\xi}_I - J_2\dot{\varphi} = 0 \tag{3.24}$$

这个表达式对单独轮子的滚动约束具有强的相似性。但是，矩阵替代了单个值，因此把全部的轮子都考虑进去了。J_2 是一个常对角 $N\times N$ 矩阵，它的实体是全部标准轮子的半径 r。对全部轮子，$J_1(\beta_s)$ 表示一个具有投影的矩阵，该投影投射到沿着它们各个轮子平面的运动上。

$$J_1(\beta_s) = \begin{bmatrix} J_{1f} \\ J_{1s}(\beta_s) \end{bmatrix} \tag{3.25}$$

注意，$J_1(\beta_s)$ 只是 β_s 的函数，不是 β_f 的函数。这是因为可操纵标准轮的方向作为时间的函数而变动，而固定标准轮的方向是恒定的。所以对所有固定标准轮，$J_1(\beta_s)$ 是一个恒定的投影矩阵，它的大小为 $N_f\times 3$。对各固定标准轮，各行由方程(3.12)的 3 -矩阵中三个项所组成；$J_1(\beta_s)$ 是一个大小为 $N_s\times 3$ 的矩阵，对各可操纵标准轮，各行由方程(3.15)的 3 -矩阵中三个项所组成。

总之，方程(3.24)表示了约束，即所有标准轮，必须根据它们沿着轮子平面的运动，围着它们的水平轴旋转适当量，使得滚动发生在地面接触点。

我们用同样的技术，把所有标准轮的滑动约束汇集成一个单独表达式，它与方程(3.13)和(3.16)具有相同的结构：

$$C_1(\beta_s)R(\theta)\,\dot{\xi}_I = 0 \tag{3.26}$$

$$C_1(\beta_s) = \begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \end{bmatrix} \tag{3.27}$$

C_{1f}和 C_{1s}分别是 $N_f \times 3$ 和 $N_s \times 3$ 的矩阵，对所有固定标准轮和可操纵的标准轮，它们的行是方程(3.13)和(3.16)中 3 -矩阵中的三个项。因此，方程(3.26)对所有的标准轮是恒定的，即正交于它们轮子平面的运动分量必定为零。这个对所有标准轮的滑动约束，如下节所述，对确定机器人底盘的总体机动性有最重大的影响。

3.2.5 举例：机器人运动学模型和约束

在 3.2.2 节中简单的差动驱动机器人的情况下，通过把各轮对机器人运动的贡献联合起来，我们提出了$\dot{\xi}_I$ 的前向运动学解答。现在我们可以利用上面提出的工具，对每一种轮子类型直接用滚动约束，构建同样的运动学表达式。我们继续把此技术再用到差动驱动的机器人中，当与 3.2.2 节结果比较时，能够验证该方法。然后，我们向三轮全向机器人迈进。

3.2.5.1 一个差动驱动机器人的实例

首先，参考方程(3.24)和(3.26)，这些方程建立了机器人运动和滚动与滑动的约束 $J_1(\beta_s)$和 $C_1(\beta_s)$，以及机器人轮子转速的关系。把这二个方程融合，我们得到以下表达式：

$$\begin{bmatrix} J_1(\beta_s) \\ C_1(\beta_s) \end{bmatrix} R(\theta)\,\dot{\xi}_I = \begin{bmatrix} J_2\dot{\varphi} \\ 0 \end{bmatrix} \tag{3.28}$$

我们再来看看图 3.3 中差动驱动机器人，可以直接从各轮的滚动约束构造$J_1(\beta_s)$和 $C_1(\beta_s)$。小脚轮是无动力的并可在任何方向自由运动，所以我们完全忽略其第三接触点。其余二个驱动轮是不可操纵的，因此，$J_1(\beta_s)$和 $C_1(\beta_s)$分别简化为 J_{1f}和 C_{1f}。为使用固定标

准轮的滚动约束方程(3.12),我们必须首先辨识各轮的 α 和 β 的值。假定,如图 3.1 所示,机器人的局部参考框架被排列得使机器人沿着 $+X_R$ 运动。在这种情况下,对右轮 $\alpha=-\pi/2,\beta=\pi$;对左轮 $\alpha=\pi/2,\beta=0$。注意到,对右轮 β 的值,必须保证正转引起在 $+X_R$ 方向运动(图 3.4)。现在我们可以利用方程(3.12)和(3.13)的矩阵项计算 J_{1f} 和 C_{1f}。因为两个固定标准轮是平行的,方程(3.13)只产生一个独立方程,而方程(3.28)给出

$$\begin{bmatrix}\begin{bmatrix}1 & 0 & l\\ 1 & 0 & -l\end{bmatrix}\\ \begin{bmatrix}0 & 1 & 0\end{bmatrix}\end{bmatrix}R(\theta)\,\dot{\xi}_I=\begin{bmatrix}J_2\dot{\varphi}\\ 0\end{bmatrix} \tag{3.29}$$

将方程(3.29)求逆,得到特定于差动驱动机器人的运动学方程:

$$\dot{\xi}_I=R(\theta)^{-1}\begin{bmatrix}1 & 0 & l\\ 1 & 0 & -l\\ 0 & 1 & 0\end{bmatrix}^{-1}\begin{bmatrix}J_2\dot{\varphi}\\ 0\end{bmatrix}=R(\theta)^{-1}\begin{bmatrix}\frac{1}{2} & \frac{1}{2} & 0\\ 0 & 0 & 1\\ \frac{1}{2l} & -\frac{1}{2l} & 0\end{bmatrix}\begin{bmatrix}J_2\dot{\varphi}\\ 0\end{bmatrix} \tag{3.30}$$

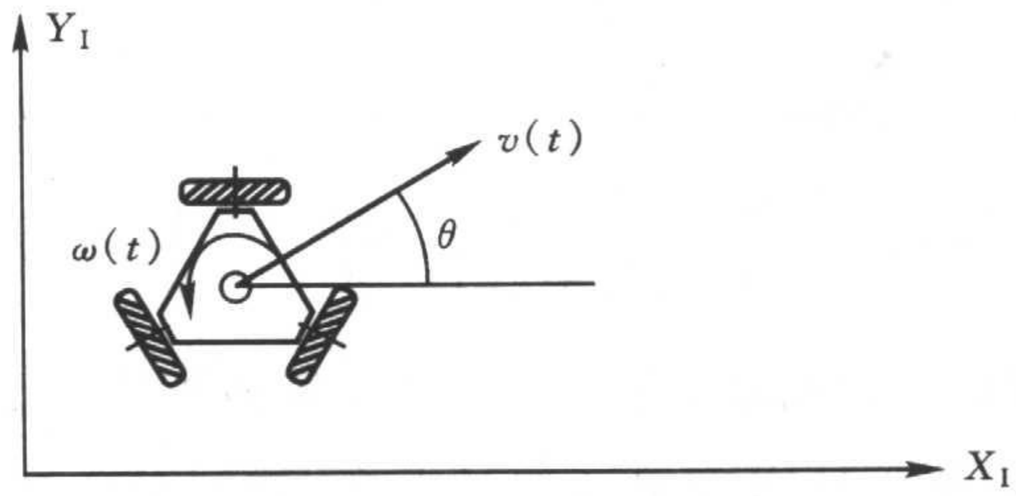
对简单的差动驱动情况,根据 3.2.2 节的手工计算,这展示了轮子滚动和滑动约束的联合描述了运动学的行为。

3.2.5.2 一个全向机器人的实例

考虑图 3.10 全向轮的机器人。该机器人有 3 个 90°瑞典轮,径向对称地安装,滚柱垂直于各主轮。

首先,我们必须在机器人上加上特定的局部参考框架。通过在机器人中心选择点 P,然后将机器人排列成与局部参考框架一致,使 X_R 与轮 2 的轴重合。图3.11表示了以这种方式安排的机器人和它的局部参考框架。

我们假定,各轮和点 P 之间的距离为 l,且所有 3 个轮子具有相同的半径 r。再次,$\dot{\xi}_I$ 的值,如在方程(3.28)一样,可以被计算为机



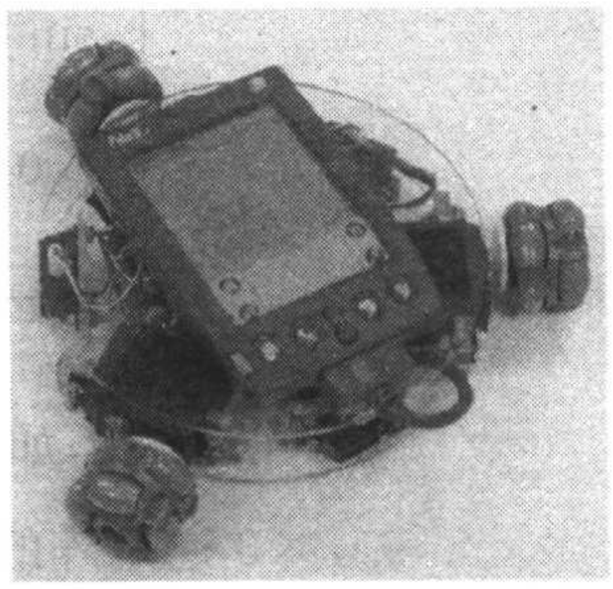


图 3.10 卡内基梅隆大学开发的三轮全向驱动机器人(www.cs.cmu.edu/~pprk)

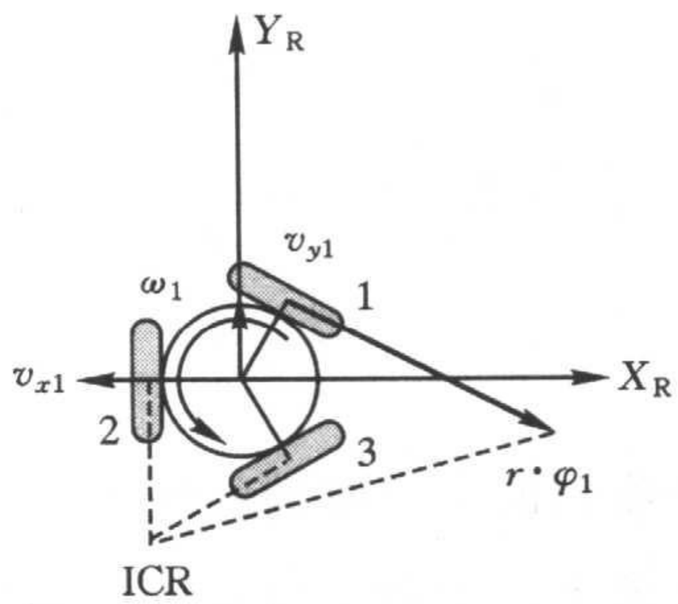


图 3.11 轮 1 的局部参考框架和详细参数

器人 3 个全向轮的滚动约束的组合。如同差动驱动机器人,因为该机器人无可操纵轮,$J_1(\beta_s)$简化为 J_{1f}:

$$\dot{\xi}_I = R(\theta)^{-1} J_{1f}^{-1} J_2 \dot{\varphi} \tag{3.31}$$

我们利用由方程(3.19)给出的瑞典轮滚动约束的矩阵元素计算 J_{1f}。但是为了使用这些值,我们必须对各轮建立 α,β,γ 的值。参见图(3.8),我们可以看到,对 90°瑞典轮,$\gamma=0$。注意到这点,马上将方程(3.19)简化成方程(3.12),即固定标准轮的滚动约束。给定局部参考框架的特别位置,对各轮计算 α 的值:$(\alpha_1=\pi/2)$,$(\alpha_2=\pi)$,$(\alpha_3=-\pi/3)$,而且对所有轮子 $\beta=0$,因为轮子都与机器人的圆形体相切。用方程(3.12)构造并简化 J_{1f},得到:

$$J_{1f}=\begin{bmatrix}\sin\frac{\pi}{3} & -\cos\frac{\pi}{3} & -l\\ 0 & -\cos\pi & -l\\ \sin-\frac{\pi}{3} & -\cos-\frac{\pi}{3} & l\end{bmatrix}=\begin{bmatrix}\frac{\sqrt{3}}{2} & -\frac{1}{2} & -l\\ 0 & 1 & -l\\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & -l\end{bmatrix} \tag{3.32}$$

再次，如方程(3.31)所示，计算$\dot{\xi}_I$要求计算逆矩阵：J_{1f}^{-1}。计算3×3逆矩阵的一种方法是用生搬硬套的办法；第二种方法，如3.2.2节所述，是计算各瑞典轮对底盘运动的贡献。我们把此过程留作学习爱好者的习题。一旦得到逆矩阵，$\dot{\xi}_I$可以是孤立的：

$$\dot{\xi}_I=R(\theta)^{-1}\begin{bmatrix}\frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}}\\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3}\\ -\frac{1}{3l} & -\frac{1}{3l} & -\frac{1}{3l}\end{bmatrix}J_2\dot{\varphi} \tag{3.33}$$

考虑一个特别的全向驱动的底盘，对所有的轮子，$l=1$和$r=1$。机器人的局部参考框架和全局参考框架是一致的，所以$\theta=0$。如果轮1,2,3分别以速度($\dot{\varphi}_1=4$)，($\dot{\varphi}_2=1$)，($\dot{\varphi}_3=2$)旋转，那么整个机器人的最终运动会是什么样呢？利用以上方程，可以容易地计算出答案：

$$\dot{\xi}_I=\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}}\\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3}\\ -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3}\end{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}4\\1\\2\end{bmatrix}$$

$$=\begin{bmatrix}\frac{2}{\sqrt{3}}\\ -\frac{4}{3}\\ -\frac{7}{3}\end{bmatrix} \tag{3.34}$$

所以，该机器人将瞬时地以正的速度沿 x 轴运动，以负的速度沿 y 轴运动，同时顺时针转动。从上面的例子我们可以看到，通过联合单个轮子的滚动约束，可以预测机器人的运动。

为了更深入一步，可以使用包含 $C_1(\beta_s)$ 的滑动约束，它使我们能计算机器人的机动性和工作空间，而不是仅仅预测它的运动。下面，我们检验使用滑动约束，有时与滚动约束结合的方法，以对机器人的机动性产生强有力的分析。

3.3 移动机器人的机动性

机器人底盘运动学的活动性是表示它在环境中直接运动的能力。限制活动性的基本约束是每一轮子必须满足它的滑动约束的规则。所以，我们可从方程(3.26)正式地推导机器人的活动性。

除了瞬时的运动学运动之外，移动机器人通过操纵可操纵的轮子，能够随时操纵它的位置，像我们将在 3.3.3 节看到那样，机器人的整个机动性就是根据标准轮运动学的滑动约束、可用的活动性以及操纵和转动可操纵的标准轮所提供的附加自由度的联合。

3.3.1 活动性的程度

方程(3.26)中加上了每一个轮子必须避免的任何横向滑动的约束。当然，对每个轮子，这个约束各自都成立。所以，对固定的和可操纵的标准轮，有可能个别地指定该约束：

$$C_{1f}R(\theta)\,\dot{\xi}_I = 0 \tag{3.35}$$

$$C_{1s}(\beta_s)R(\theta)\,\dot{\xi}_I = 0 \tag{3.36}$$

为满足这两个约束，运动向量 $R(\theta)\dot{\xi}_I$ 必须属于投影矩阵 $C_1(\beta_s)$ 的**零空间**，它是 C_{1f} 和 C_{1s} 的简单联合。在数学上，$C_1(\beta_s)$ 的零空间是空间 N，使得对任何 N 中向量 n，$C_1(\beta_s)n=0$。如果遵守运动学的约

束，则机器人的运动必定总是在该空间 N 内。在几何上，利用机器人**转动的瞬时中心**(instantaneous center of rotation，ICR)的概念，同样可以证明运动学的约束(方程(3.36))。

来看一个单独的标准轮，它迫于滑动的约束没有横向运动。在几何上，可以经过它的水平轴，垂直于轮子平面，画一条**零运动的直线**(图 3.12)来证明。

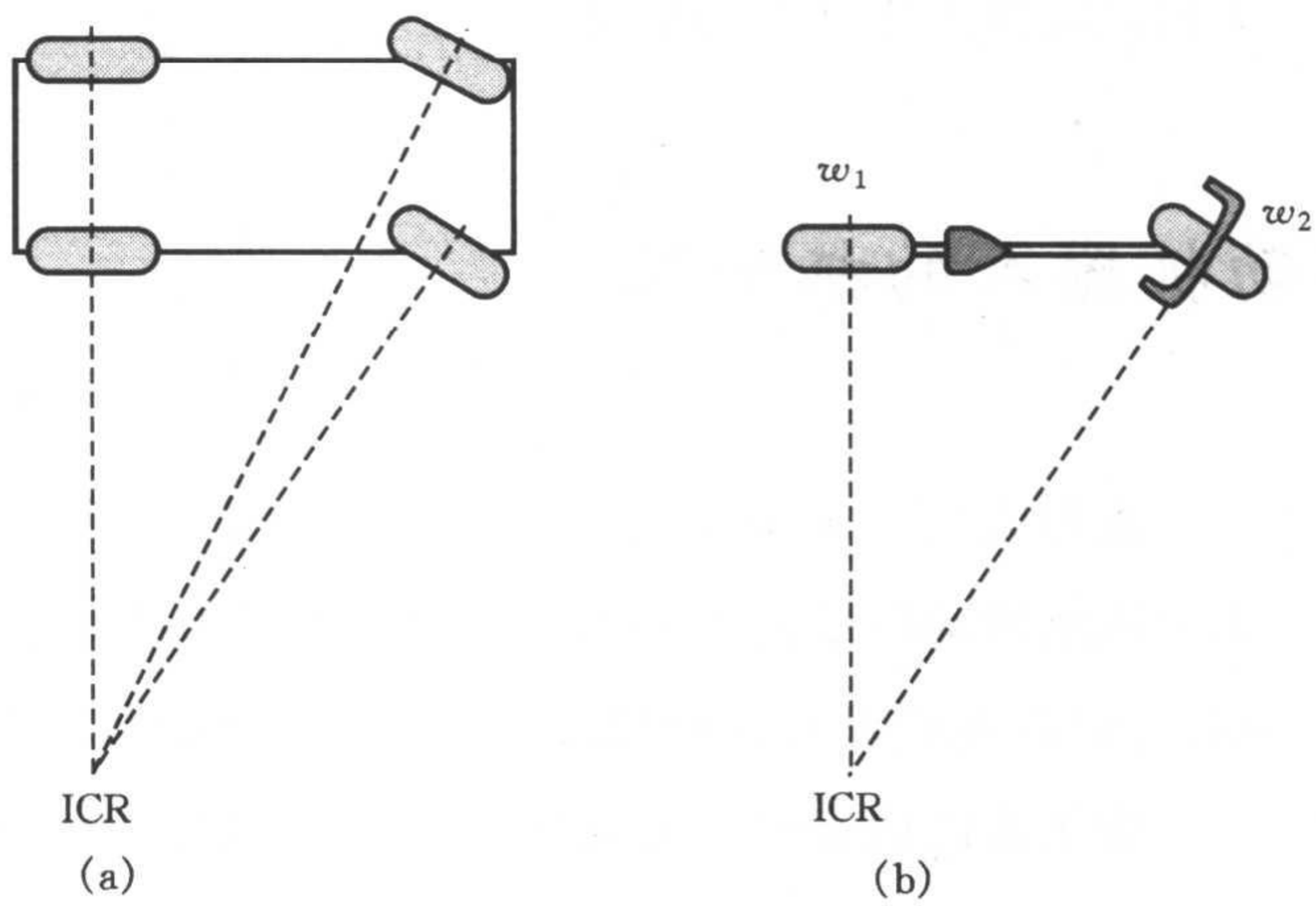


图 3.12 (a)具有拟汽车操纵的四轮；(b)自行车

在任何给定时刻，沿着零运动直线的轮子运动必为零。换句话说，轮子必沿着半径为 R 的某个圆瞬时地运动，使得那个圆的中心处在零运动直线上。该中心点称为转动的瞬时中心，可以位于沿零运动直线的任何地方。当 R 为无穷大时，轮子按直线运动。

像图 3.12(a)中 Ackerman 车辆那样的机器人可以有好几个轮子，但必须总是有一个单独的 ICR，因为所有它的零运动直线在一个单独点相交，将 ICR 放在该交点，机器人运动存在一个单独的解。

这个 ICR 的几何结构，显示了机器人的活动性如何是机器人运动上约束数目的函数而不是轮子数目的函数。在图 3.12(b)中，所示的自行车有两个轮子，w_1 和 w_2，各轮提供一个约束，或一个零运动线。合在一起，两个约束产生一个单独的点作为唯一的 ICR 剩余

解。这是因为两个约束是独立的，因此各约束限制了整个机器人的运动。

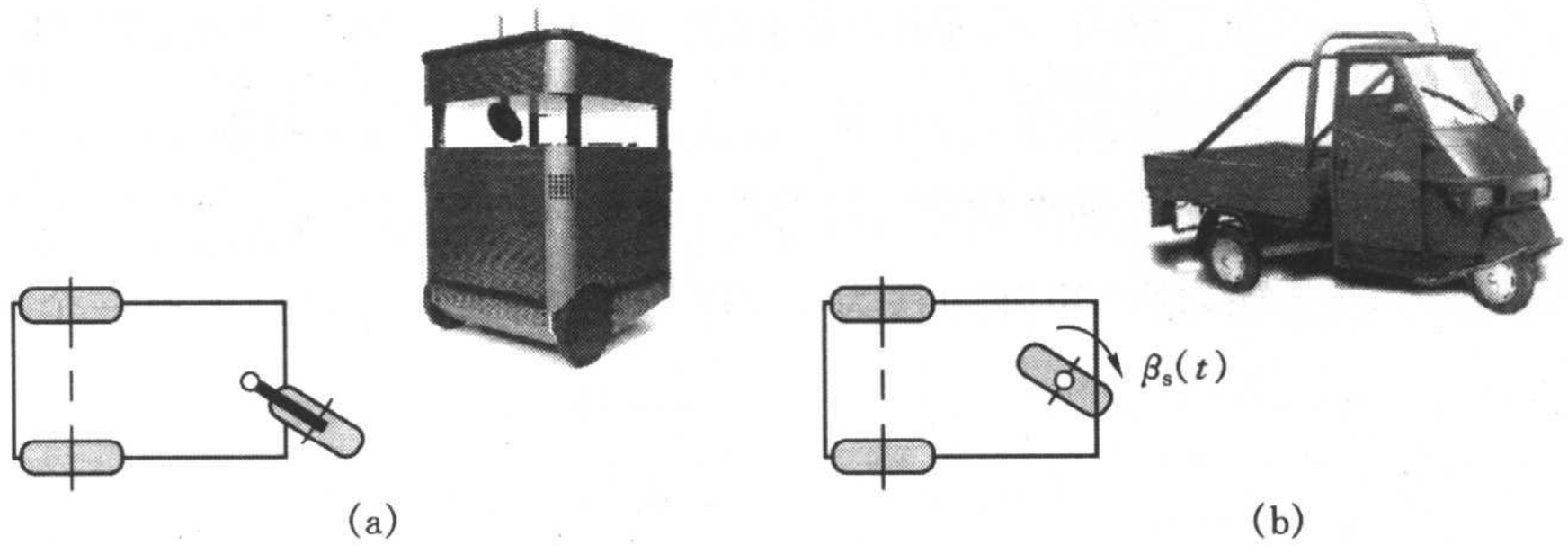


图 3.13 (a)具有两个各自动力化轮子和一个小脚轮的差动驱动机器人，例如 EPFL 的 Pygmalion 机器人；(b)具有两个固定轮和一个可操纵轮的三轮车，如 Piaggio 的机械手

但是在图 3.13(b)差动驱动的机器人情况下，两个轮子沿相同的水平轴排列，所以，ICR 被限制在沿一条直线展开，而不是在一个特定的点。事实上，第二个轮子在机器人运动上没有加附加的运动学约束，因为它的零运动线是与第一个轮子一样的。这样，虽然自行车和差动驱动的底盘有同样数目的非全向轮，前者有两个独立的运动学约束，而后者只有一个。

图 3.12(a)的 Ackerman 车辆展示了另一种方法，在那里，轮子对机器人的运动学也许不能提供一个独立的约束。该车辆有两个可操纵的标准轮。只给定其中一个可操纵轮的瞬时位置和固定的后轮的位置，只存在 ICR 的单独解。第二个可操纵轮的位置绝对地受 ICR 限制。所以，它对机器人的运动不提供独立的约束。

所以，机器人的底盘运动学是由所有标准轮引起的**独立**约束集合的函数。独立的数学上的解释涉及到矩阵的**秩**(rank)。记得，一个矩阵的秩就是独立的行或列的最少数目。方程(3.26)代表了由运动机器人所加的全部的滑动约束。所以，$[C_1(\beta_s)]$的秩就是独立约束的数目。

独立约束的数目越多，$[C_1(\beta_s)]$的秩就越大，机器人的活动性就

越受约束。例如，考虑一个具有单个固定轮的机器人，记住，我们只考虑标准轮。该机器人可以是单轮的，或者它可以有几个瑞典轮，然而它确有一个固定标准轮。轮子所处位置由相对于机器人局部参考框架的参数 α,β,l 指定。$C_1(\beta_s)$ 由 C_{1f} 和 C_{1s} 组成。然而，因为没有可操纵的标准轮，C_{1s} 是空的，故 $C_1(\beta_s)$ 只包含 C_{1f}。由于没有固定标准轮，这个矩阵的秩为1，所以该机器人在活动性上有一个单独的独立约束：

$$C_1(\beta_s) = C_{1f} = [\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta] \tag{3.37}$$

现在让我们加一个附加的固定标准轮，把第二轮子像原始轮子一样，限制成与相同的水平轴排列一致。不失一般性，我们可以把 P 点放到两个轮子中心之间的中点。对轮 w_1，给定 α_1,β_1,l_1；对轮 w_2，给定 α_2,β_2,l_2，它在几何上保持 $\{(l_1=l_2),(\beta_1=\beta_2=0),(\alpha_1+\pi=\alpha_2)\}$。所以，在这种情况下，矩阵 $C_1(\beta_s)$ 有两个约束，而秩为1：

$$C_1(\beta_s) = C_{1f} = \begin{bmatrix} \cos(\alpha_1) & \sin(\alpha_1) & 0 \\ \cos(\alpha_1+\pi) & \sin(\alpha_1+\pi) & 0 \end{bmatrix} \tag{3.38}$$

换一种方法，考虑将 w_2 放到 w_1 的轮平面但具有相同方向的情况，如同在前向位置具有锁定操纵的自行车一样。我们再次把点 P 放在两轮中心的中点，且把轮子定向，使它们位于轴 x_1。这个几何特征意味着 $\{(l_1=l_2),(\beta_1=\beta_2=\pi/2),(\alpha_1=0),(\alpha_2=\pi)\}$，所以，矩阵 $C_1(\beta_s)$ 保持两个约束，且秩为2：

$$\begin{aligned} C_1(\beta_s) = C_{1f} &= \begin{bmatrix} \cos(\pi/2) & \sin(\pi/2) & l_1\sin(\pi/2) \\ \cos(3\pi/2) & \sin(3\pi/2) & l_1\sin(\pi/2) \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & l_1 \\ 0 & -1 & l_1 \end{bmatrix} \end{aligned} \tag{3.39}$$

一般而言，如果 $[C_{1f}]$ 的秩大于1，则在最好情况下，车辆只能沿着一个圆或一条直线行走。这种结构意味着，由于固定标准轮不共享同一个旋转的水平轴，机器人有两个或更多的独立约束。因为这种结构在平面中只有活动性的退化形式，在本章的其余部分我们不

考虑它们。然而，请注意某些退化结构，如四轮滑动/制动操纵系统在某种环境中是有用的，如在松散泥土和沙地上，即使它们不满足滑动约束。不用惊奇，对这种破坏滑动约束必须付出的代价是基于里程的航位推测，随之变成低准确度，并剧烈地降低了动力效率。

一般机器人会有零个或多个固定标准轮和可操纵标准轮。所以我们对任何机器人辨识秩值的可能范围：$0 \leqslant rank[C_1(\beta_s)] \leqslant 3$。考虑 $rank[C_1(\beta_s)]=0$ 的情况。这只可能发生在 $C_1(\beta_s)$ 中有零独立运动学约束，在该情况下，在机器人框架上既不装固定的也不装可操纵的标准轮：$N_f=N_s=0$。

考虑另外极端，$rank[C_1(\beta_s)]=3$。这是最大可能的秩，因为沿着 3 个自由度指定运动学约束（即约束矩阵为 3 列宽）。所以，不存在 3 个以上的独立约束。事实上，当 $rank[C_1(\beta_s)]=3$ 时，机器人在任何方向是完全受约束的，所以是退化的，因为它完全不可能在平面中运动。

现在我们准备好从形式上定义机器人的**活动性程度** δ_m：

$$\delta_m = dimN[C_1(\beta_s)] = 3 - rank[C_1(\beta_s)] \tag{3.40}$$

矩阵 $C_1(\beta_s)$ 零空间的维数（$dimN$）是通过改变轮子的速度，可以被立即操纵的机器人底盘的自由度数目的一个度量。所以，从逻辑上讲，δ_m 必在 0～3 之间变化。

让我们考虑一个普通的差动驱动的底盘。在这个机器人上，有两个共享公共水平轴的两个固定标准轮。如上述讨论那样，第二个轮子对系统不加独立的运动学约束。所以，$rank[C_1(\beta_s)]=1$ 和 $\delta_m=2$。这与直觉一致：**简单地通过操纵轮子的速度**，差动驱动的机器人既可以控制方向的变化速率，又可以控制它的向前/向后的速度。换句话说，它的 ICR 被限制在位于从它的轮子水平轴扩展的无限直线上。

相反，考虑一个自行车的底盘，这个结构由一个固定标准轮和一个可操纵的标准轮所组成。在这种情况下，各轮对 $C_1(\beta_s)$ 提供一个

独立的滑动约束。所以，$\delta_m=1$。注意，自行车拥有的非全向轮的总数和差动驱动的底盘相同，而且其中一个轮子确实是可操纵的，但它的活动性程度少1。从反应来看，这是合适的。通过直接对轮子速度的操纵，自行车只能对它的向前/向后速度进行控制；只有通过操纵自行车才能改变它的ICR。

正如猜想的那样，根据方程(3.40)，只有由全向轮所组成的任何机器人，诸如瑞典轮或球形轮，会有最大的活动性程度，$\delta_m=3$。这种机器人可以直接操纵所有的3个自由度。

3.3.2 可操纵度

上面定义的活动性程度，根据轮子速度变化将可控的自由度进行了量化。操纵也可以对机器人底盘的姿态 ξ 有最终的影响，虽然影响是间接的。因为改变可操纵标准轮的角度后，机器人必须因操纵角度的变化而运动，对姿态有影响。

如活动性一样，当定义**可操纵度** δ_s 时，我们关心有关独立地可控的操纵参数的数目：

$$\delta_s = rank[C_{1s}(\beta_s)] \tag{3.41}$$

回忆一下，在活动性情况下，增加 $C_1(\beta_s)$ 的秩，意味着更多的约束，因而是移动较少的系统。在可操纵性的情况下，增加 $C_1(\beta_s)$ 的秩，意味着更多的操纵自由度，从而有更大的最终机动性。因为 $C_1(\beta_s)$ 包含 $C_{1s}(\beta_s)$，这意指可操纵的标准轮可以既减少活动性又增加可操纵性。在任何时刻，它的特殊方位加了一个运动学约束，但它改变那个方位的能力可以产生附加的轨迹。

可以指定 δ_s 的范围为：$0\leqslant\delta_s\leqslant2$。$\delta_s=0$ 的情况说明机器人没有可操纵的标准轮，$N_s=0$。当机器人的结构包含一个或多个可操纵标准轮时，$\delta_s=1$ 的情况是最普通的。

例如，考虑一辆普通汽车，在该情况下，$N_f=2$ 和 $N_s=2$。但固定轮共享一个公共轴，所以 $rank[C_{1f}]=1$。固定轮和任何一个可操

纵轮限制 ICR 成为沿着从后轴延伸的直线的一个点。所以，第二个可操纵轮不能再加任何独立的运动学约束，故 $rank[C_{1s}(\beta_s)]=1$，在此情况下，$\delta_m=1$ 和 $\delta_s=1$。

$\delta_s=2$ 的情况只可能在机器人无固定标准轮时出现，即 $N_f=0$，才可能。在这些环境下，有可能制作具有两个各别可操纵轮的底盘，像一个伪自行车(或双操纵)。这里两个轮子都是可操纵的。因此，调整一个轮子的方向就把 ICR 限制为一条直线，而第二个轮子可以把 ICR 限制在沿着那直线的任何点。有趣的是，这意味着 $\delta_s=2$ 时，说明机器人可以把 ICR 放到地平面上任何地方。

3.3.3 机器人的机动性

机器人可以操纵的总的自由度，称为**机动程度** δ_M，它可以根据活动性和可操纵性很容易地被定义：

$$\delta_M = \delta_m + \delta_s \tag{3.42}$$

所以，机动性包括通过轮子的速度，机器人直接操纵的自由度和通过改变操纵的配置和运动，间接操纵的自由度两个方面。根据前面几节的研究，人们可画出轮子配置的基本类型，如图 3.14 所示。

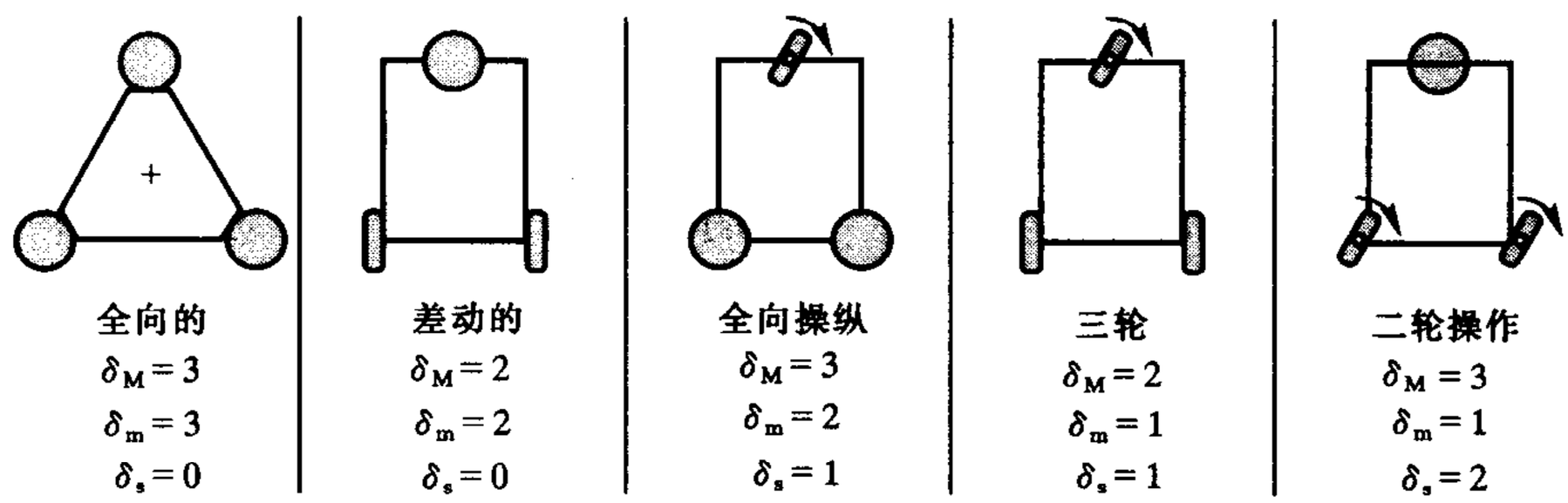


图 3.14 三轮配置的五个基本类型。球形轮可以由小脚轮或瑞典轮代替而不影响机动性。在第 2 章有具有不同数目轮子的更多的配置

注意，具有相同 δ_M 的两个机器人不必是等效的。例如，差动驱动和三轮几何特征(图 3.13)具有相等的机动性 $\delta_M=2$。在差动驱动中，所有的机动性是直接的活动性的结果，因为 $\delta_m=2$ 和 $\delta_s=0$。在

三轮情况下，机动性也由操纵引起：$\delta_m=1$ 和 $\delta_s=1$。这些结构不论哪一个都不允许 ICR 放在平面的任何地方。在这两种情况下，ICR 必须相对于机器人参考框架，位于预定的直线上。在差动驱动的情况下，该直线从两个固定标准轮的公共轴延伸，用差动轮的速度将 ICR 点设置在该直线上。在三轮情况下，该直线从固定轮的共享公共轴延伸，用可操纵轮沿该直线设置 ICR。

更一般地，对任何 $\delta_M=2$ 的机器人，ICR 总是被限制在位于一条直线上，而对 $\delta_M=3$ 的机器人，ICR 可以被设置在平面上的任何点。

最后一个例子将展示我们上面已开发的工具的使用。为室内移动机器人学研究的一个普通机器人的结构是**同步驱动**结构（图 2.22）。这种机器人有 2 个电机和 3 个锁定在一起的轮子。一个电机为转动所有 3 个轮子提供动力，而第二个电机为操纵所有 3 个轮子提供动力。在三轮同步系统机器人中，$N_f=0$ 和 $N_s=3$。所以，可以用 $rank[C_{1s}(\beta_s)]$来确定 δ_m 和 δ_s。3 个轮子并不共享一个公共轴，其中 2 个提供独立的滑动约束。为了使运动成为可能，第三个必须依赖于这 2 个的约束。因此，$rank[C_{1s}(\beta_s)]=2$ 和 $\delta_m=1$。这在直觉上是正确的。方向固定的同步驱动的机器人只有 1 个自由度，它构成了在一条直线上的前后行走。

然而，当考虑 δ_s 时，它产生了一个有意义的应用。根据方程（3.41），机器人应该有 $\delta_s=2$。的确，对于具有同步驱动几何结构的三轮操纵机器人，这是正确的。但是我们还有附加的信息：在一个同步驱动结构中，单个电机用皮带操纵所有 3 个轮子。所以，虽然是理想的，如果轮子都是独立可操纵的，则系统会实现 $\delta_s=2$。在同步驱动的情况下，驱动系统进一步限制了运动学，使得在现实中 $\delta_s=1$。最后，我们可以根据这些数值计算机动性：对同步驱动的机器人而言 $\delta_M=2$。

这个结果说明，一个同步驱动的机器人总共只能操纵 2 个自由度。事实上，如果读者仔细想一想同步驱动机器人的轮子结构，就会

清楚，这里没有方法改变底盘的方向。只有底盘的 $x-y$ 位置可以被操纵。所以，同步驱动机器人确实只有 2 个自由度，它与我们数学上的结论一致。

3.4 移动机器人工作空间

对于一个机器人来说，机动性等效于它的控制自由度。但是，机器人是处于某种环境的，因而下一个问题是把我们的分析放到环境之中。我们关心机器人用它可控制的自由度在环境中定位它本身的方法。例如，考虑 Ackerman 车辆或汽车，对这种车辆而言，控制的自由度总数是 $\delta_M=2$。一个为操纵；另一个为驱动轮的执行。但在它的环境中，车辆总的自由度是什么呢？事实上，它是 3 个。汽车可以在平面上将它自己定位于任何点 x,y，并具有角度 θ。

因此，辨识可能配置的机器人的空间是重要的。因为，令人惊奇的是，它可以超过 δ_M。除了**工作空间**，我们关心有关机器人如何能在不同配置环境之间进行运动：它可以跟踪的路径类型是什么样的？进一步，穿过这个配置空间，它的可能轨迹是什么样的？在讨论的其余部分，我们把重点从内部的运动学细节，如轮子，转向机器人底盘的姿态和底盘的自由度。记住这一点，现在让我们把机器人放到它的工作空间的背景中。

3.4.1 自由度

在定义机器人的工作空间时，首先检验**可容许的速度空间**是有用的。给定机器人的运动学约束，它的速度空间描述了机器人可以控制的机器人运动的独立分量。例如，独轮车的速度空间可以用两个轴来表示：一个表示独轮车的瞬时前向速度；另一个表示独轮车瞬时方向变化 $\dot{\theta}$。

在机器人的速度空间中，维数就是独立地可达到的速度的数目。这也被称为**可微的自由度**(differentiable degrees of freedom，DDOF)。机器人的DDOF常常等于活动性程度δ_m。例如，自行车具有以下的机动性：$\delta_M=\delta_m+\delta_s=1+1=2$。自行车的DDOF确为1。

与自行车相反，我们考虑一个全向机器人，即具有3个瑞典轮的机器人。我们知道，在这情况下，没有一个标准轮。故$\delta_M=\delta_m+\delta_s=3+0=3$。所以全向机器人有3个微分自由度。这是合适的，因为这种机器人没有运动学运动的约束，它能够独立地设置全部3个姿态变量：$\dot{x}$，$\dot{y}$，$\dot{\theta}$。

给定自行车和全向机器人之间在DDOF中的差别，我们考虑在各配置的工作空间中总的自由度。全向机器人在它的环境中可以获得任何姿态(x,y,θ)，这通过同时地直接获得所有3个轴的目标位置是可以做到的，因为DDOF=3。显然，它有一个DOF=3的工作空间。

自行车能否在它的环境中获得任何姿态(x,y,θ)？它可以做到，但是达到某些目标点也许需要比等效的全向机器人更多的时间和能量。例如，如果自行车结构必须横向移动1 m，最简单的成功技巧类似于汽车的平行停车，会涉及到螺旋式或前后式运动。尽管如此，自行车可以到达任何(x,y,θ)，所以自行车的工作空间同样有DOF=3。

显然，在工作中存在一个不等式关系：$\text{DDOF}\leqslant\delta_M\leqslant\text{DOF}$。虽然机器人的工作空间是一个重要的属性，从上面的例子中可以明白，特殊的可用路径，对机器人也是事关重要的。正如工作空间DOF支配机器人的能力获得不同姿态一样，机器人的DDOF也支配它的能力获得不同的路径。

3.4.2 完整机器人

在机器人学的学界中，当描述移动机器人的路径空间时，通常使

用“**完整**”这一概念。术语**完整**对几个数学领域有广泛的可用性，包括微分方程、函数和约束表达式。在移动机器人学中，该术语特别地当作机器人底盘的运动学约束。一个**完整机器人**是没有非完整运动学约束的机器人；相反，一个非完整机器人是具有一个或多个非完整运动学约束的机器人。

完整的运动学约束可以被表示为只是位置变量的显函数。例如，在具有单个固定标准轮的移动机器人的情况下，完整的运动学约束只用 $\alpha_1, \beta_1, l_1, r_1, \varphi_1, x, y, \theta$ 就可以表达。这种约束不能用这些变量的微分，诸如 $\dot{\varphi}$ 或 $\dot{\xi}$。**非完整运动学约束**需要微分关系，如位置变量的微分。而且它不能只根据位置变量对它积分以提供约束。根据后者的观点，非完整系统常常被称为**非可积**系统。

考虑固定标准轮的滑动约束：

$$[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta]R(\theta)\,\dot{\xi}_I = 0 \tag{3.43}$$

这个约束必须用机器人的移动 $\dot{\xi}$，而不是姿态 ξ，因为这一点，它限制了垂直于轮子平面的机器人的移动为零。这个约束是不可积分的，它明显地依赖于机器人的运动。所以，滑动约束是一个非完整约束。考虑具有一个固定标准轮和一个可操纵标准轮的自行车结构，因为对这种机器人，固定轮的滑动约束将是强制的。我们可以得出结论，自行车是一个非完整的机器人。

但如果有人把自行车操纵系统锁定，使得它变成两个固定轮，各具独立但平行的轴。我们知道，对这样系统，$\delta_M = 1$。它是非完整的吗？由于滑动和滚动约束，虽然它也许不这样表露出来，但锁定的自行车实际上是完整的。考虑锁定的自行车的工作空间，它由一条单独的无限长直线组成，沿着此直线自行车可以运动(假定操纵被锁定笔直向前)。为方程简单起见，假定该无限长直线与全局参考框架中 X_I 相一致，且 $\{\beta_{1,2} = \pi/2, \alpha_1 = 0, \alpha_2 = \pi\}$。在这种情况下，两个轮子的约束可以用一个机器人姿态上相等的约束全集：$\{y=0, \theta=0\}$ 来替代。这就消去了相应于两个轮子的滑动约束的两个非完整约束。

留下的非完整运动学约束只是各轮的滚动约束：

$$[-\sin(\alpha+\beta) \quad \cos(\alpha+\beta) \quad l\cos\beta]R(\theta)\,\dot{\xi}_I + r\dot{\varphi} = 0 \tag{3.44}$$

对各轮，要求该约束将轮子旋转的速度和沿着轮子平面投影的运动速度建立关系。但在我们锁定自行车的情况下，给定原点的轮子初始的转动位置 φ_0，我们可以用一个量代替这个约束，该量把直线上的位置 x 直接与轮子的转动角度 φ 关联起来：$\varphi=(x/r)+\varphi_0$。

锁定的自行车是第一种类型的完整机器人的一个例子——这里确实存在约束，但都是完整的运动学约束。对所有 $\delta_M<3$ 的所有完整形机器人，都是这样的情况。当没有运动学约束时，即 $N_f=0$ 和 $N_s=0$，就存在第二种类型的完整形机器人。因为没有运动学约束，故也没有非完整的运动学约束，所有这种机器人总是完整的。对所有 $\delta_M=3$ 的完整形机器人都是这样的情况。

另一种描述完整机器人的方法是基于机器人的微分自由度和它的工作空间自由度之间的关系：**一个机器人是完整的，当且仅当** DDOF=DOF。直觉上，这是因为只有通过非完整约束（由可操纵或固定标准轮所加），机器人才能获得一个自由度超过微分自由度的工作空间，DOF>DDOF。例子包括差动驱动和二轮/三轮结构。

在移动机器人中，实用的底盘，通常必须在维数为3的工作空间中实现姿态。所以一般地，对底盘我们需要 DOF=3。但是，围绕障碍物操纵而不影响方向的“完整”能力，以及在跟踪任意路径时的跟踪目标的能力，都是重要的附加考虑。为了这些缘故，对移动机器人学最有关系的完整的特殊形式是 DDOF=DOF=3。我们将这类机器人结构定义为全向的：**全向的机器人**是一个 DDOF=3 的完整机器人。

3.4.3 路径和轨迹的考虑

在移动机器人学中，我们不仅关心有关机器人达到所需的最后配置的能力，也关心它是**如何**获得成功的。现在考虑机器人跟踪路

径的能力问题：在最佳的情况下，机器人应该能够跟踪穿过它姿态工作空间的任何路径。显然地，任何全向机器人能够这样做是因为它在三维工作空间是完整的。遗憾的是，全向机器人必须使用无约束轮，轮子的选择仅限于瑞典轮、小脚轮和球形轮。这些轮子还未被具体用在结构设计中，那将导致工作量巨大的地面和悬挂物的清除。虽然从路径空间的观点看，它们是强有力的，但是它们比固定和可操纵轮的通用性少得多，主要是因为它们的设计和制作多少有点复杂和昂贵。

另外，非完整约束可大大改善运动的稳定性。考虑一个在恒定直径曲线上高速驱动的全向车辆，在这样运动时，车辆易受到不可忽略的向心力，这个将车辆推出曲线的横向力必须被全向轮的电机转矩所抵消。在电机或控制失败的情况下，车辆会被甩出曲线外。然而对一个具有运动学约束的拟汽车的机器人来说，横向力通过滑动约束被动地予以抵消，减轻了对电机转矩的要求。

回忆一下利用标准轮的一个高机动性的较早的例子：自行车上两个轮子都是可操纵的，通常称作**双操纵**。这个车辆获得可操纵性的自由度为 2，产生高度的机动性：$\delta_M = \delta_m + \delta_s = 1 + 2 = 3$。有意义的是，该结构是非完整的，但在 DOF＝3 的工作空间中具有高度的机动性。

机动性的结果 $\delta_M = 3$，意味双操纵通过适当地操纵它的两个轮子，可以选择任何 ICR。那么如何把它与全向机器人比较呢？在平面中，操纵它的 ICR 的能力意味双操纵可以在它的工作空间中跟踪任何路径。更为一般地说，$\delta_M = 3$ 的任何机器人可以在工作空间中，从它的初始姿态到它的最后姿态，跟踪任何路径。一个全向机器人也可以在它的工作空间中跟踪任何路径，而且，毋用惊奇，因为在全向机器人中 $\delta_m = 3$，它必遵顺 $\delta_M = 3$。

但是，在由操纵所授予的自由度和由轮子速度直接控制所授予的自由度之间仍有差别。这个差别，清楚地是在**轨迹**背景上而不是

在路径上。一个轨迹除了占有一个附加维数——时间外，它与一条路径是一样的。所以，对地面上一个全向机器人而言，一条路径一般地表示了通过一个 3D 姿态空间的踪迹；对相同的机器人，一个轨迹表示了加上时间，经过 4D 姿态空间的踪迹。

例如，考虑一个目标轨迹，在此轨迹中，机器人沿 X_I 轴以 1 m/s 恒速运动 1 秒，然后也在 1 秒中逆时针改变方向 90°，然后平行于 Y_I 轴运动 1 秒。设计好的 3 秒轨迹如图 3.15 所示，它利用了相对于时间的 x, y, θ 图。

全向机器人能实现这个轨迹吗？我们假定机器人的每个轮子能达到某任意的、有限的速度。为了简单起见，我们进一步假定，加速度是无限的，即它可不花时间地达到任意的期望速度。在这些假定下，全向机器人的确可以跟踪图 3.15 的轨迹。例如，第 1 秒和第 2 秒之间的过渡只涉及轮子速度的改变。

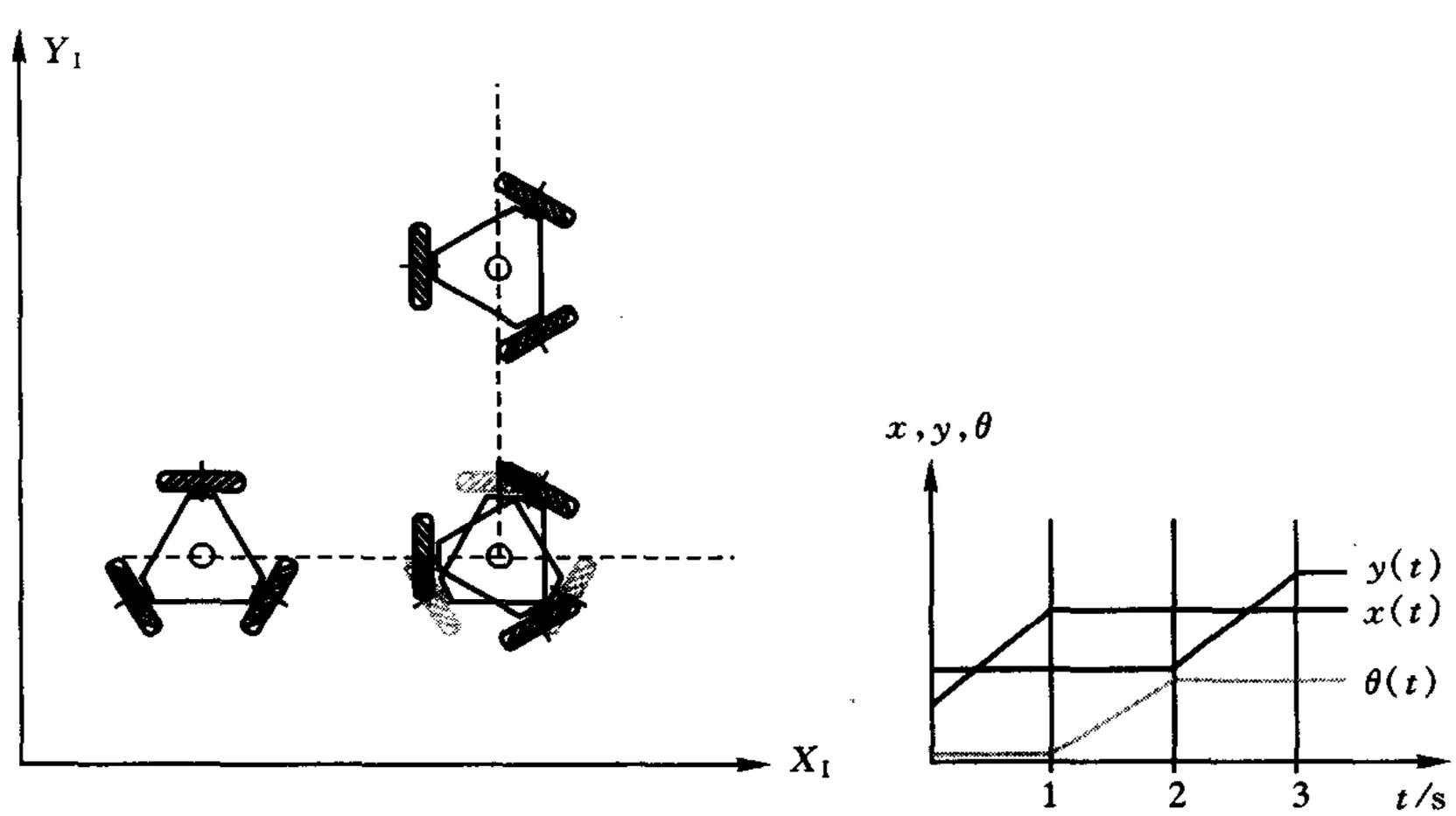


图 3.15 全向机器人的机器人轨迹的例子：沿 X_I 轴以恒速 1 m/s 运动 1 秒；逆时针改变方向 90° 1 秒钟；沿 Y_I 轴以恒速 1 m/s 运动 1 秒

因为双操纵有 $\delta_M = 3$，它必定能跟踪将轨迹投影到无时间的工作空间所产生的路径。然而，它不能跟踪这个 4D 的轨迹，即使操纵速度是有限且任意的。虽然双操纵会有能力瞬时地改变操纵速度，

但在开始改变机器人底盘方向之前，必须等待可操纵轮的角度变到期望的位置。简言之，双操纵需要改变成内部的自由度，而且因为这些改变要花时间，任意轨迹是达不到的。图 3.16 表示了双操纵可以达到的最相似的轨迹。与期望的运动的 3 个阶段不同，该轨迹有 5 个阶段。

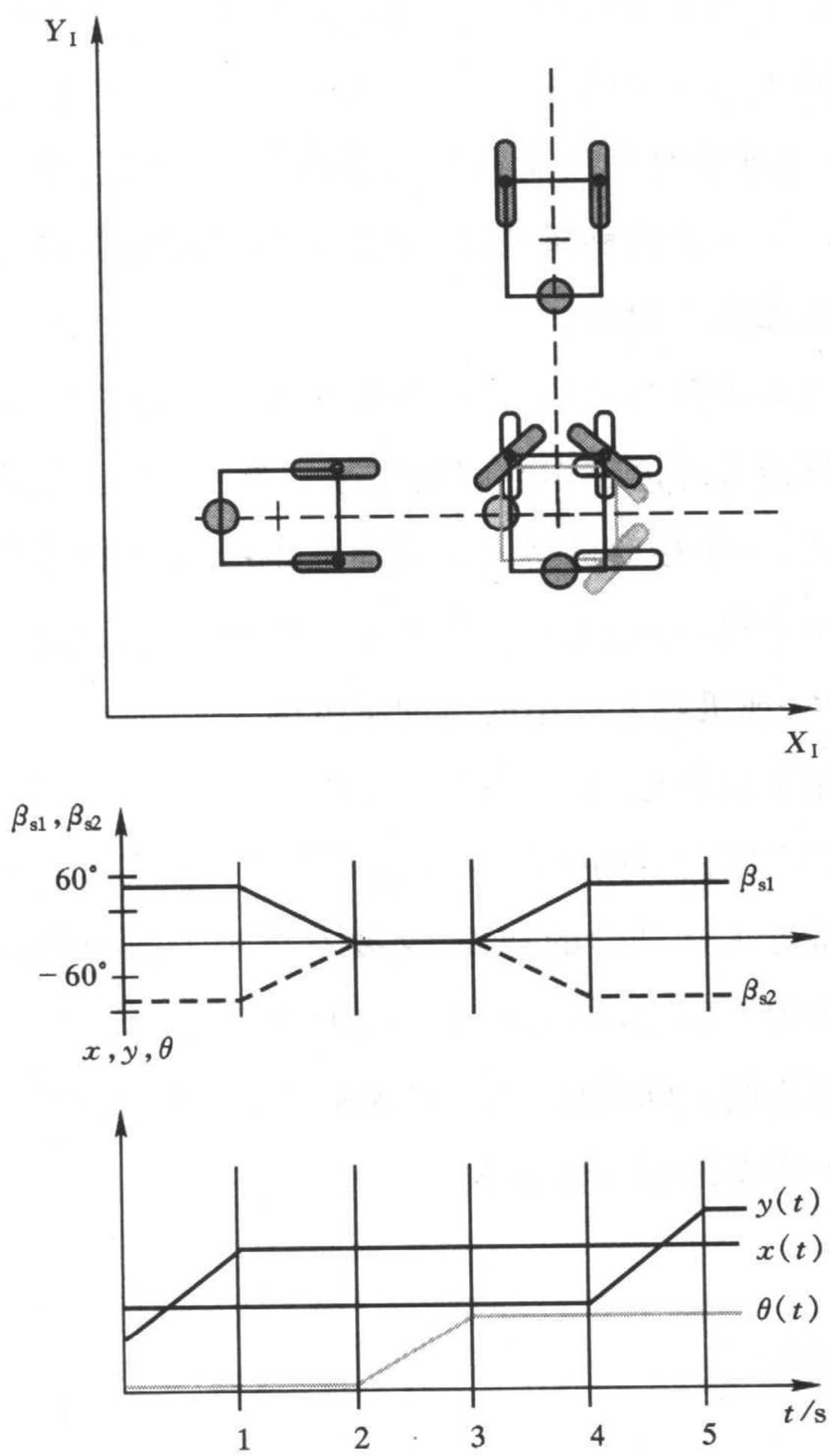


图 3.16 具有操纵轮与图 3.15 相似的机器人轨迹的实例：沿 X_I 轴以恒速 1 m/s运动 1 秒；分别旋转操纵轮 −50°/50°，逆时针改变方向 90°1 秒钟；分别旋转操纵轮 50°/−50°，沿 Y_I 轴以恒速 1 m/s 运动 1 秒

3.5 基本运动学之外

以上移动机器人运动学的讨论只是极其丰富命题的一个介绍。当我们也考虑速度和力时，这在高速移动机器人情况下特别地需要，于是除了运动学约束之外，必须表达动力学约束。而且，许多移动机器人，诸如坦克型底盘和四轮滑动/制动系统，不服从上面的运动学模型。在分析这种系统时，通常需要显式地将机器人和地平面之间的粘摩擦进行建模。

更有重要意义的是，移动机器人的运动学分析，提供了关于那种移动机器人理论工作空间的结果。然而，为了在该工作空间中有效地运动，一个移动机器人必须有它的自由度的适当激励。这个问题称为动力化(摩托化)，它要求进一步分析力，且必须主动地供给这个力，以实现机器人可用的运动学范围。

除了动力化之外，还存在能控性问题：在什么条件下，移动机器人能在有限的时间从初始姿态行走到目标姿态？回答这个问题既需要有机器人运动学的知识，又要有可用于激励移动机器人的控制系统的知识。所以，移动机器人控制回到了设计一个现实世界的控制算法这样的实际问题，算法利用应用领域所要求的轨迹，可以一个姿态一个姿态地驱动机器人。

3.6 运动控制

如上所述，对非完整系统，运动控制(运动学控制)也许并不是一个容易的任务。但不同的研究群体已研究了这个问题，例如参考文献[8,39,52,53,137]，而且已经有了移动机器人运动控制的某些合

适的解决方案。

3.6.1 开环控制

运动控制器的目标是跟踪一条轨迹，该轨迹由它的位置或速度轮廓描述成为时间的函数。这是通过将轨迹(路径)分割成形状清晰的被定义的运动区段来完成的，例如直线或圆弧段。因此，控制问题是根据直线和圆弧段，预先计算平滑的轨迹，驱动机器人从初始位置

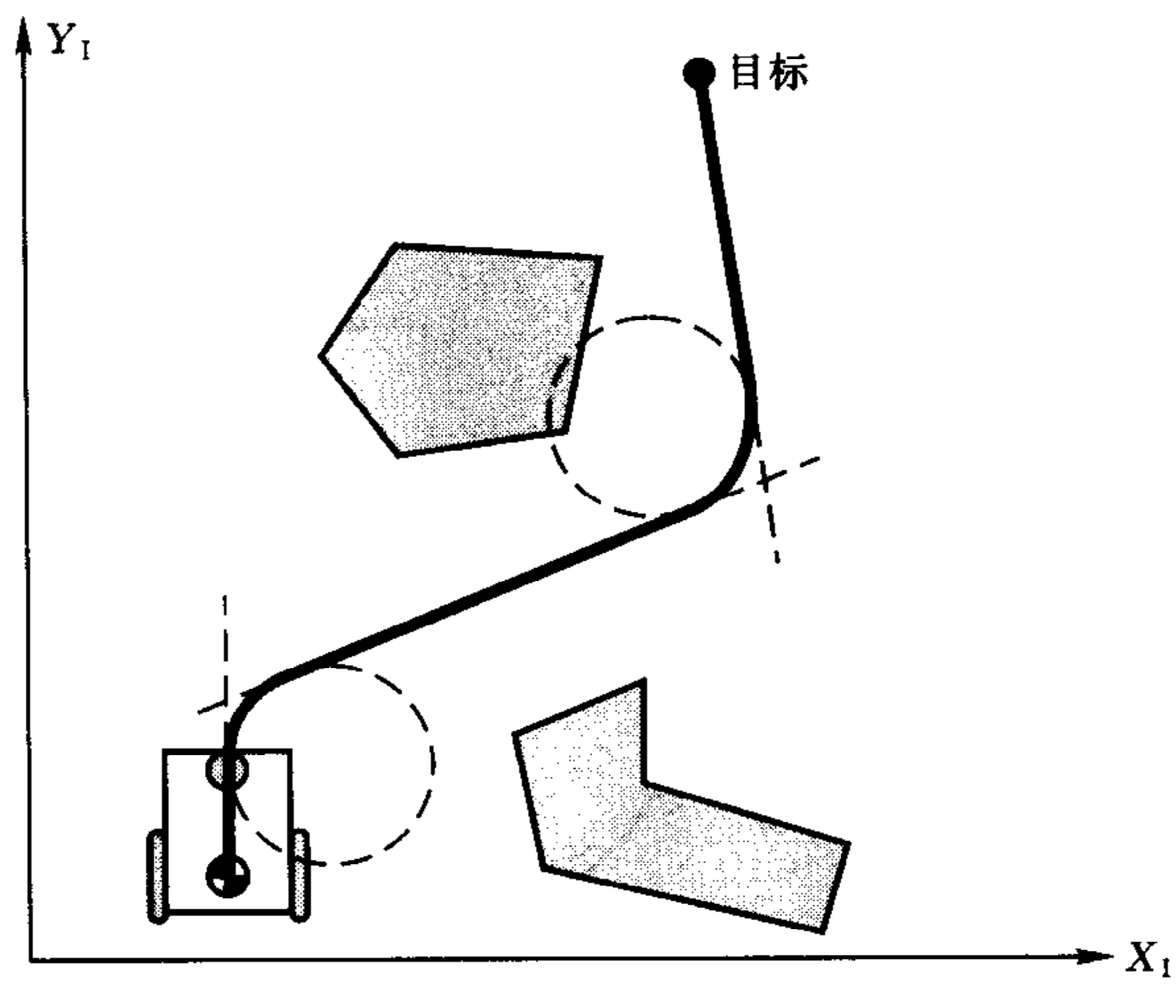


图 3.17 根据直线和圆轨迹段的运动机器人的开环控制

走到最终位置(图 3.17)。这个方法可以被当作开环运动控制(轨迹跟踪)，因为所测量的机器人位置不被反馈当作速度或位置控制。它有几个缺点：

- 如果必须考虑机器人的速度和加速度的所有限制和约束，预先计算可行的轨迹根本不是一个容易的任务。
- 如果环境发生动态的改变，机器人不会自动地适应或修正轨迹。
- 最终的轨迹常常不是平滑的。因为对大多数常用的区段(例如，直线和圆的一部分)，从一个轨迹段到另一个轨迹段是不

平滑的。这意味着在机器人的加速度中，存在不连续性。

3.6.2 反馈控制

在移动机器人的运动控制中，更为适宜的方法是使用一个状态反馈控制器。有了这种控制器，机器人的路径规划任务被简化为在所要求的路径中间点，设置位置(子目标)。在3.6.2.1节中，为稳定的差动驱动移动机器人的反馈控制，提出了一个有用的解决方案。它非常类似于参考文献[39,100]中提出的控制器。其他的方法可在参考文献[8,52,53,137]中找到。

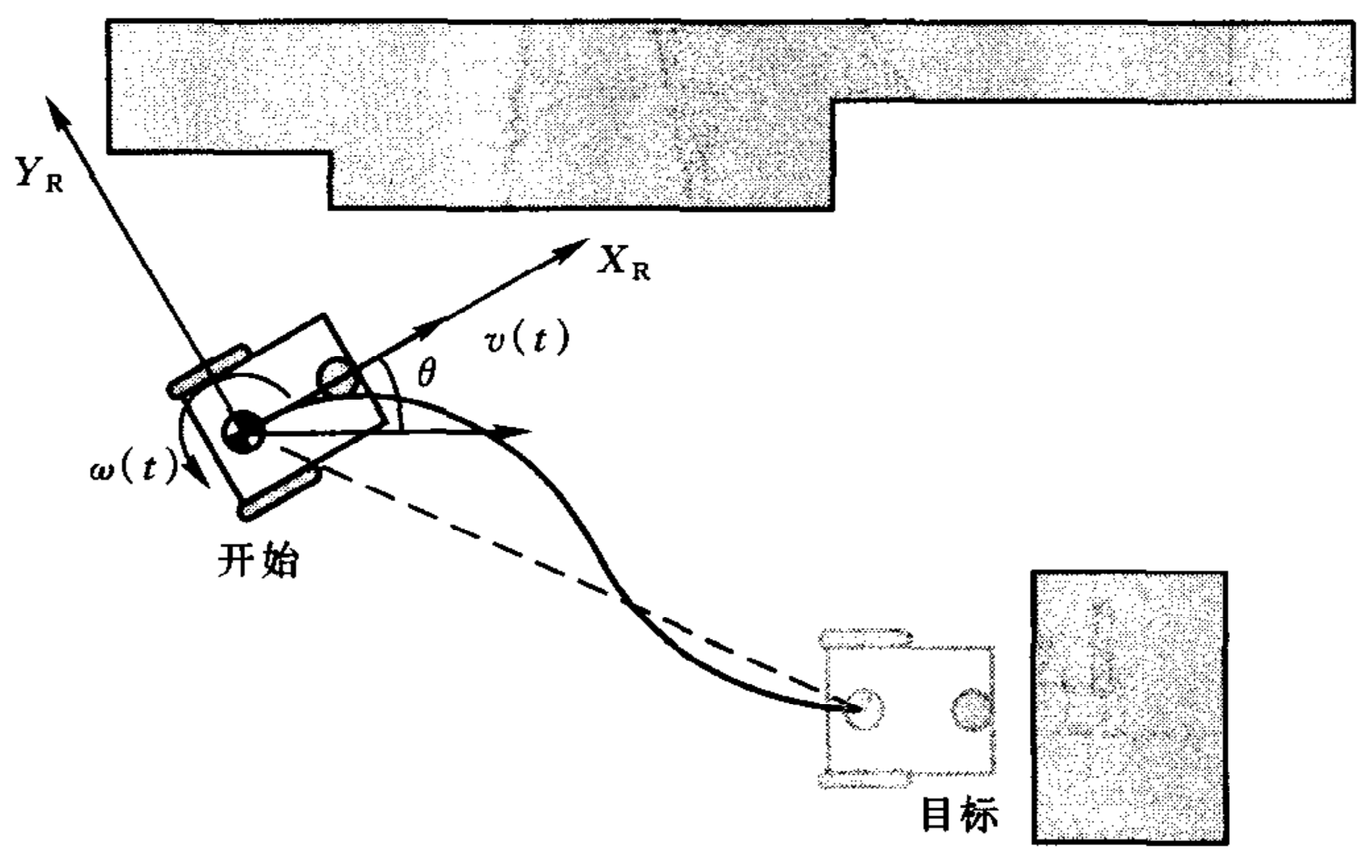


图3.18 移动机器人反馈控制的典型情况

3.6.2.1 问提陈述

考虑图3.18所示的情况，机器人具有任意的位置和方向，以及一个预定的目标位置和方向，在机器人参考框架$\{X_R, Y_R, \theta\}$中，所给定的实际姿态误差向量为$e={}^R[x, y, \theta]^T$，x，y和θ是机器人的目标坐标。

控制器设计的任务是寻求一个控制矩阵，如果它存在的话，

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix}; \quad k_{ij} = k(t, e) \tag{3.45}$$

使得 $v(t)$ 和 $w(t)$ 的控制

$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = K \cdot e = K \, {}^{R}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{3.46}$$

驱使误差 e 趋向零*。

$$\lim_{t \to \infty} e(t) = 0 \tag{3.47}$$

3.6.2.2 运动学模型

不失一般性，我们假定目标是在惯性框架的原点(图 3.19)。在下面，位置向量总是被表示在惯性框架中。

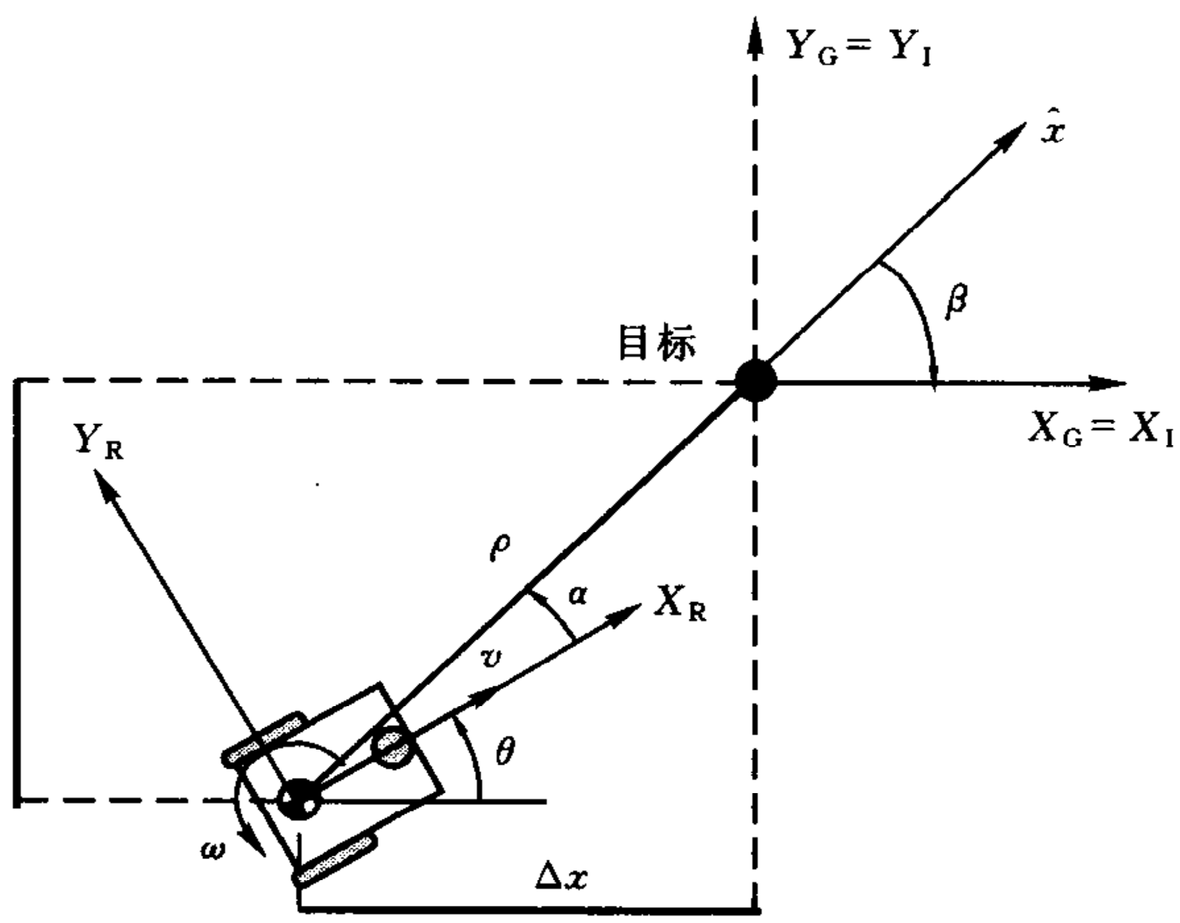


图 3.19 机器人的运动学和它感兴趣的框架

在惯性框架 $\{X_I, Y_I, \theta\}$ 中，所描述的差动驱动移动机器人的运动学，给定如下：

$${}^{I}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3.48}$$

* 记住，由于非完整的约束，$v(t)$ 总是朝向机器人参考框架的 X_R 方向。

式中$\dot{x}$和$\dot{y}$是在惯性框架 X_I 和 Y_I 方向中的直线速度。

令 α 表示机器人参考框架 X_R 轴和连接轮子轴中心与最终位置的向量$\dot{x}$之间的角度。如果 $\alpha \in I_1$，这里

$$I_1 = \left(-\frac{\pi}{2}, \frac{\pi}{2}\right] \tag{3.49}$$

那么考虑坐标转换到原点在目标位置的极坐标

$$\rho = \sqrt{\Delta x^2 + \Delta y^2} \tag{3.50}$$

$$\alpha = -\theta + \arctan2(\Delta y, \Delta x) \tag{3.51}$$

$$\beta = -\theta - \alpha \tag{3.52}$$

这样利用矩阵方程，在新的极坐标中，得到一个系统的描述

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos\alpha & 0 \\ \dfrac{\sin\alpha}{\rho} & -1 \\ -\dfrac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3.53}$$

式中 ρ 是机器人轮轴的中心与目标位置之间的距离，θ 表示机器人参考框架 X_R 轴和与最后位置有关的 X_I 轴之间的角度，v 和 ω 分别为切向速度和角速度。

另一方面，如果 $\alpha \in I_2$，式中

$$I_2 = (-\pi, -\pi/2] \cup (\pi/2, \pi] \tag{3.54}$$

通过设置 $v = -v$，重新定义机器人的前向方向，我们得到由矩阵形式描述的系统：

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 \\ -\dfrac{\sin\alpha}{\rho} & 1 \\ \dfrac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3.55}$$

3.6.2.3 对极坐标中运动学模型的评注

- 坐标变换不是被定义在 $x = y = 0$，因为在这样的点，变换的雅可比矩阵的行列式是不定的，即是无界的。

- 对 $\alpha \in I_1$，机器人的前向方向指向目标；对 $\alpha \in I_2$，它是相反方向。
- 在它的初始配置中，通过适当地定义机器人的前向方向，常常有可能在 $t=0$ 时，有 $\alpha \in I_1$。然而，这并不意味对所有时间，α 都保持在 I_1。因此，为了避免在到达目标时机器人改变方向，如果可能，有必要以这样的方式确定控制器：只要 $\alpha(0) \in I_1$，则对所有 t，$\alpha \in I_1$。对反方向，同样可用（见以下稳定性问题）。

3.6.2.4 控制律

现在必须设计控制信号 v 和 ω，以把机器人从它的实际位置，比如 $(\rho_0, \alpha_0, \beta_0)$，驱动到目标位置。显然，方程(3.53)在 $\rho=0$ 出现不连续，但 Brockett 定理不妨碍平滑稳定性。

如果我们现在考虑线性控制律

$$v = k_\rho \rho \tag{3.56}$$

$$\omega = k_\alpha \alpha + k_\beta \beta \tag{3.57}$$

我们用方程(3.53)得到下式描述的闭环系统

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho \rho \cos\alpha \\ k_\rho \sin\alpha - k_\alpha \alpha - k_\beta \beta \\ -k_\rho \sin\alpha \end{bmatrix} \tag{3.58}$$

系统在 $\rho=0$ 没有任何奇异，并在 $(\rho, \alpha, \beta) = (0,0,0)$ 有一个唯一的平衡点。因此，它会驱动机器人到该点，这就是目标位置。

- 在笛卡儿坐标系，控制律（方程(3.57)）产生了在 $x=y=0$ 是不定的方程。
- 要知道这样事实：角度 α 和 β 经常被表达在范围 $(-\pi, \pi)$ 中。
- 观察到控制信号 v 经常有一个恒定的符号，即只要 $\alpha(0) \in I_1$，它是正的；其他情况它总是负的。这说明机器人执行它的停车操作总是在单个方向，且无逆向运动。

在图 3.20 中，当机器人起初处在 x, y 平面的一个圆上时，可得

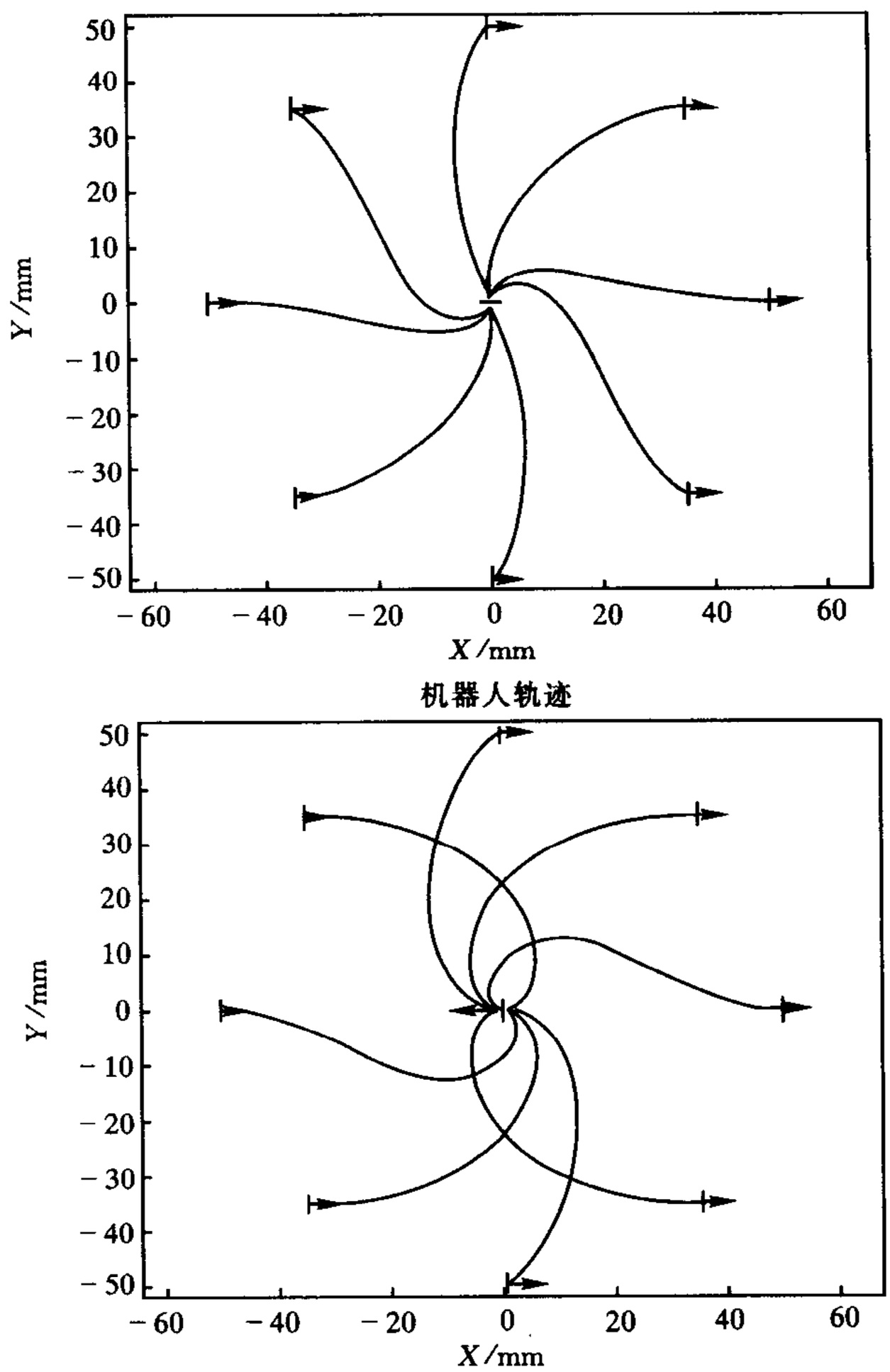


图 3.20 当机器人初始在 x,y 平面单位圆上时的最终路径

到最终的路径。所有的运动具有趋向目标在中心的平滑的轨迹。对该仿真，控制参数被设置为

$$k=(k_\rho,k_\alpha,k_\beta)=(3,8,-1.5) \tag{3.59}$$

3.6.2.5 局部稳定性问题

进一步可以证明，闭环控制系统（方程(3.58)）局部指数地稳定的，如果

$$k_\rho>0;\ k_\beta<0;\ k_\alpha-k_\rho>0 \tag{3.60}$$

证明：

围绕平衡点位置($\cos x=1,\sin x=x$)线性化，方程(3.58)可以被写成：

$$\begin{bmatrix}\dot{\rho}\\ \dot{\alpha}\\ \dot{\beta}\end{bmatrix}=\begin{bmatrix}-k_\rho & 0 & 0\\ 0 & -(k_\alpha-k_\rho) & -k_\beta\\ 0 & -k_\rho & 0\end{bmatrix}\begin{bmatrix}\rho\\ \alpha\\ \beta\end{bmatrix}, \tag{3.61}$$

因此，它是局部指数地稳定，如果矩阵的特征值

$$A=\begin{bmatrix}-k_\rho & 0 & 0\\ 0 & -(k_\alpha-k_\rho) & -k_\beta\\ 0 & -k_\rho & 0\end{bmatrix} \tag{3.62}$$

全部有负的实部。矩阵 A 的特征多项式为

$$(\lambda+k_\rho)(\lambda^2+\lambda(k_\alpha-k_\rho)-k_\rho k_\beta) \tag{3.63}$$

所有的根有负的实部，如果

$$k_\rho>0;\ -k_\beta>0;\ k_\alpha-k_\rho>0 \tag{3.64}$$

证毕。

为了对鲁棒位置控制，也可建议用强的稳定性条件，它保证在机器人到达它的目标时，不改变方向：

$$k_\rho>0;\ k_\beta<0;\ k_\alpha+\frac{5}{3}k_\beta-\frac{2}{\pi}k_\rho>0 \tag{3.65}$$

这分别说明，只要 $\alpha(0)\in I_1$，对所有 $t,\alpha\in I_1$；只要 $\alpha(0)\in I_2$，对所有 $t,\alpha\in I_2$。这个强稳定性条件在应用中还未被证明。

第 4 章　感知

任何种类的自主系统，其最重要的任务之一是获取关于其环境的知识。这是用不同的传感器感受测量并从那些测量中提取有意义信息而实现的。

本章我们提出在移动机器人中最常用的传感器，而后讨论从传感器提取信息的策略。许多用于移动机器人的更详细资料，可参阅 H. R. Everett 所著的综合性书籍《移动机器人传感器》[15]。

4.1　移动机器人的传感器

用在移动机器人的传感器种类广泛(图 4.1)。有些传感器只用于测量简单的值，像机器人电子器件内部温度或电机转速。而其他更复杂的传感器可以用来获取关于机器人环境的信息甚或直接测量机器人的全局位置。本章主要着重于提取有关机器人环境信息的传感器。因为机器人四处移动，它常常碰见未预料的环境特征，所以这种感知特别重要。我们从传感器功能的分类开始，然后，在提出描述传感器特性指标的基本工具之后，对所选的传感器进行详细描述。

4.1.1　传感器分类

我们用两个重要的功能轴：**本体感受的/外感受的**和**被动/主动**将传感器分类。

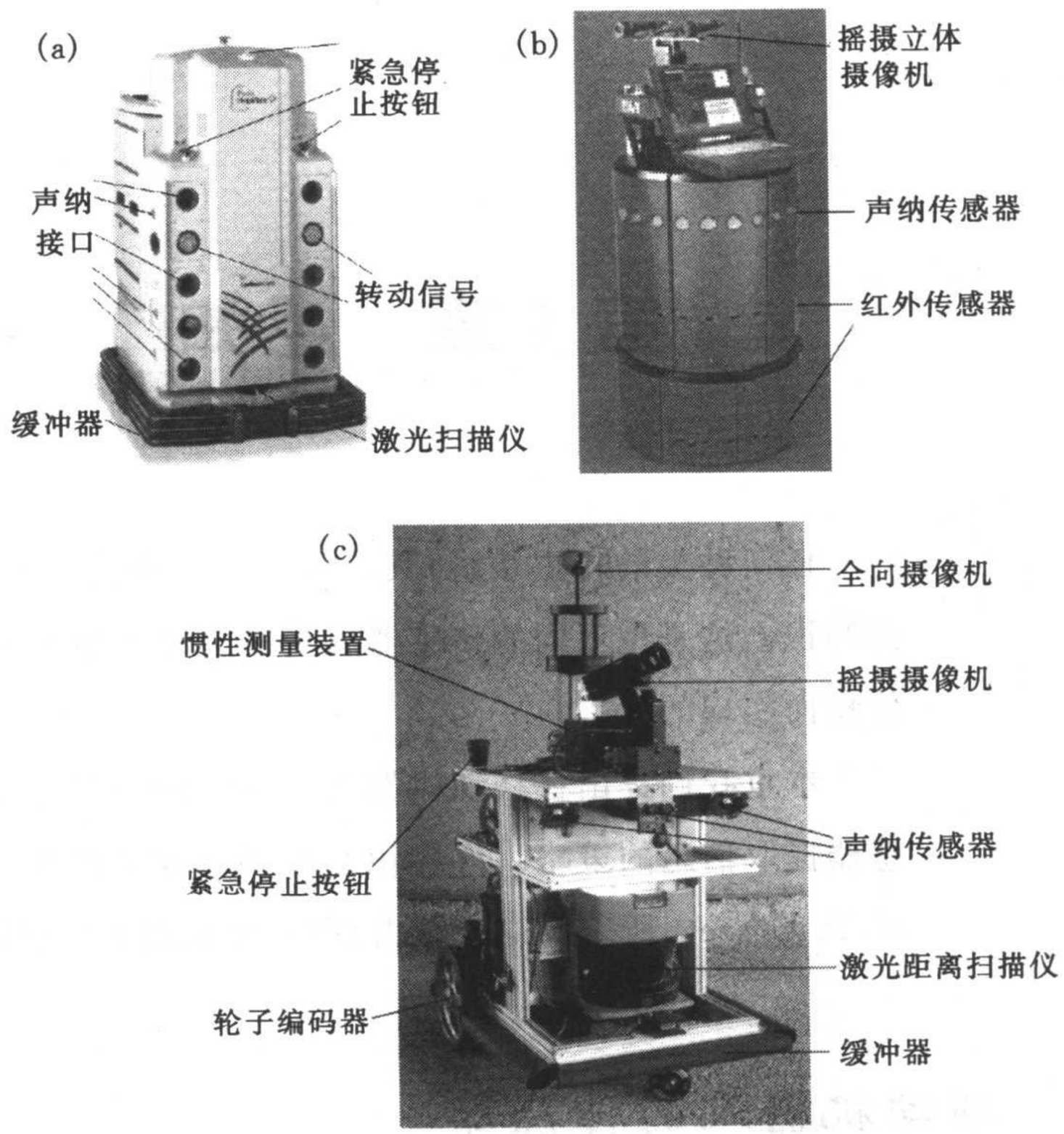


图 4.1 具有多传感器机器人的例子:(a) TRC 公司的 HelpMate;(b) 现实环境接口 B12;(c) BlueBotics SA 的 BIBA 机器人

本体感受传感器测量系统(机器人)的内部值。例如,电机速度、轮子负载、机器人手臂关节的角度、电池电压等。

外感受传感器从机器人的环境获取信息。如距离测量、亮度、声音幅度等。因此,为了提取有意义的环境特征,机器人要解释外感受传感器的测量。

被动传感器测量进入传感器的周围环境的能量。被动传感器的例子包括温度探测器、话筒、CCD 和 CMOS 摄像机。

主动传感器发射能量到环境,然后测量环境的反应。因为主动传感器可以支配与环境(受更多约束的)的交互,它们常常具有卓越

的特性指标。然而，主动传感器引入了几个风险：发出的能量可能影响传感器力图测量的真正特征。而且，主动传感器可能在它的信号和不受它控制的信号之间遭受干扰。例如，附近其他机器人或同一机器人上相似传感器发射的信号，会影响其最终的测量。主动传感器的例子包括轮子正交编码器、超声传感器和激光测距仪。

表 4.1 提供了移动机器人应用中最有用的传感器分类，本章讨论其中最有意义的传感器。

表 4.1 中，传感器按复杂性上升次序和技术成熟性下降次序排列。触觉传感器和本体感受式传感器实际上对所有移动机器人都是关键的，而且了解充分，便于实现。例如，可以购买商用正交编码器用作齿轮马达装配的零件。在另一极端，利用一个或多个 CCD/CMOS 摄像机的视觉解释，它提供了广泛的潜在机能，从避障和定位到人脸的辨识。然而，提供视觉机能的商用传感器现在刚开始出现。

4.1.2 表征传感器的特性指标

本章我们描述的传感器，其特性指标的特征变化很大。有些传感器在控制良好的实验室环境中，具有极高的准确度，但当现实环境变动时，就抵挡不住误差。其他一些传感器在种类广泛的环境中，提供狭窄、高精度的数据。为了将这些特性指标的特征定量化，我们首先正式定义传感器特性指标的术语，为统领本章的其余部分，这是有价值的。

4.1.2.1 基本传感器响应的额定值

在实验室环境中，可以定量地标定许多传感器的特征。当传感器被安置在现实环境的机器人时，这种特性指标额定值必须是处在最佳情景下。虽然如此，但仍是有用的。

表 4.1 移动机器人应用中的传感器分类

一般分类 (典型使用)	传感器 传感器系统	PC 或 EC	A 或 P
触觉传感器 (物理接触或接近检测;安全切换)	接触开关,减震器 光栅栏 非接触逼近传感器	EC EC EC	P A A
轮子/电机传感器 (轮子/电机速度和位置)	刷式编码器 电位计 同步机,分解器 光学编码器 磁性编码器 电感编码器 电容编码器	PC PC PC PC PC PC PC	P P A A A A A
导向传感器 (相对于固定参考框架的机器人方向)	罗盘 陀螺仪 倾角罗盘	EC PC EC	P P A/P
地面信标 (在固定参考框架定位)	GPS 有源光学或 RF 信标 有源超声信标 反射式信标	EC EC EC EC	A A A A
主动测距 (反射、飞越时间和几何三角测量)	反射传感器 超声传感器 激光测距仪 光学三角测量(1D) 结构光(2D)	EC EC EC EC EC	A A A A A
运动/速度传感器 (相对于固定或移动物体的速度)	多普勒雷达 多普勒声音	EC EC	A A
基于视觉传感器 (视觉测距、全像分析、分割、对象辨识)	CCD/CMOS 摄像机 视觉测距包 物体跟踪包	EC	P

A:主动;P:被动;P/A:被动/主动;PC:本体感受;EC:外感受。

在维持正常传感器操作的同时，**动态范围**用于测量传感器的输入上、下限范围。形式上，动态范围是最大输入值与最小可测输入值之比。因为这个原始比率可能不太合适，所以常常用**分贝**计量，它被计算为动态范围的普通对数的 10 倍。然而，在计算分贝时，存在潜在的混淆，它照例应该是测量功率比，诸如瓦特或马力。假如传感器测量电机电流，并且可从最小 1 mA 到 20 A 取值。这个电流传感器的动态范围定义为：

$$10\log[\frac{20}{0.001}]=43\ \text{dB} \tag{4.1}$$

现在假定你有一个测量机器人电池电压的电压传感器，测量 1 mV 到 20 V 任何电压值，电压不是功率的单位，但电压的平方正比于功率，所以我们用 20 而不用 10。

$$20\log[\frac{20}{0.001}]=86\ \text{dB} \tag{4.2}$$

在移动机器人应用中，范围也是一个重要的额定值，因为机器人的传感器经常运行在输入值超过它们工作范围的环境中。在这种情况下，关键在于了解传感器将如何响应。例如，光学测距仪有一个最小的操作范围，当对象比该最小值更靠近而进行测量时，它会产生虚假数据。

分辨率是可以被传感器检测的两个值之间的最小差别。通常，传感器动态范围的下限等于它的分辨率。然而，在数字传感器的情况下，不必如此。例如，假定有一个传感器，它测量电压，执行模-数(A/D)转换，输出为线性地相应于 0～5 V 之间的 8 位被转换值。如果传感器是真正线性的，则它有 2^8-1 个总输出值，或 5 V(255)的分辨率为 20 mV。

当输入信号变化时，**线性度**是支配传感器输出信号行为的重要测量。一个线性响应说明，如果两个输入 x 和 y 产生两个输出 $f(x)$ 和 $f(y)$，则对任何数值 a 和 b，$f(ax+by)=af(x)+bf(y)$。这意味着传感器输入/输出响应图是简单的直线。

带宽或频率用于测量速度，按此速度，传感器可以提供读数（数据）流。形式上，每秒测量数目定义为传感器的频率，单位为 Hz。由于经越环境而运动的动态性，移动机器人通常被障碍检测传感器的带宽限制在最大速度上。因此，增加测距带宽以及基于视觉的传感器已成为机器人学领域高优先级的目标。

4.1.2.2 现场传感器特性指标

在实验室环境中，可以合理地测量上述传感器的特征，并有信心外推到现实环境配置中的特性指标。然而，不深入了解所有环境特征和所考虑传感器之间的复杂交互，就不能可靠地获取许多重要的测量。这与最复杂的传感器关系最密切，包括主动测距传感器和视觉解释传感器。

灵敏度本身是一个希望的特征。这是目标输入信号增量变化引起输出信号变化的一种度量。形式上，灵敏度是输出变化与输入变化之比。然而遗憾的是，外感受传感器的灵敏度常常被不期望的灵敏度和特性指标（与其他环境参数相关）所混淆。

交叉灵敏度是正交于传感器目标参数的环境参数的灵敏度技术用语。例如，磁通闸门罗盘可以对磁北展示高的灵敏度，因而用作移动机器人导航。然而，罗盘也对含铁的建筑材料展示高的灵敏度，以至它的交叉灵敏度常常使传感器在某些室内环境中变得无用。传感器的高交叉灵敏度一般是不希望的，特别当它不能被建模时。

在某指定的运行背景内，**误差**被定义为传感器输出测量和被测的真正值之间的差。给定真值 v 和测量值 m，我们可以定义误差为误差$=m-v$。

准确度定义为传感器测量和真值之间的符合程度，通常表达成真值的比例部分（比如，97.5%准确度）。因此，小的误差与高的准确度相关，反之也然：

$$\left(\text{准确度}=1-\frac{|\text{误差}|}{v}\right) \tag{4.3}$$

当然，获得真正的真值可能是困难的或者是不可能的，所以建立传感器准确度的可信特征可能是有问题的。而且，区分两种不同误差源之间的差别是重要的。

系统误差是由理论上可建模的因素或过程造成的。所以这些误差是确定性的(即可预测的)。激光测距仪不良的标度、走廊地面不可建模的倾斜以及由于以前碰撞引起的弯曲的立体摄像机头，都可能是传感器系统误差的起因。

随机误差既不能用复杂的模型预测，也不能用更精密的传感器机构来减小。这些误差只能用概率予以描述(即随机性的)。彩色摄像机的色调、不稳定性、虚假的测距误差以及摄像机黑电平噪声都是随机误差的例证。

精确度常常和准确度相混淆，现在我们有工具清楚地区分这两个术语。直觉上，高精确度关系到传感器测量结果的重复性。例如，一个获取相同环境状态的多个读数的传感器，如果它产生相同结果，则它具有高的精确度。在另外的例子中，这个接受相同环境状态数据的传感器的多个拷贝，如果它们的输出一致，则它具有高的精确度。实际上精确度与相对于被测真值的传感器输出准确度没有任何关系。假定传感器的**随机误差**用某平均值 μ 和标准偏差 σ 予以表征，那么精确度形式上的定义是传感器输出范围和标准偏差之比：

$$\text{精确度}=\frac{\text{范围}}{\sigma} \tag{4.4}$$

注意，仅仅是 σ，不是 μ 对精确度有影响。相反，平均误差 μ 直接正比于总的传感器误差，并反比于传感器准确度。

4.1.2.3 表征传感器的误差：移动机器人学的挑战

移动机器人严重地依赖于外感受式的传感器。许多这些传感器集中在机器人的中心任务上：获取机器人紧接邻域中物体的信息，使得它可以了解它周围环境。当然，这些围绕机器人的“物体”都是从

它局部参考框架的角度检测到的。因为我们所研究的系统是移动的，它们不断移动的位置和它们的运动对整个传感器的行为有重大的影响。在本节中，借助前面讨论过的术语，我们来描述移动机器人的误差与前一节刻画的理想情景是如何地大相径庭。

系统和随机误差的模糊性 主动测距传感器容易有失效的模式，它很大程度上由传感器和环境目标的特定相对位置所引发。例如，声纳传感器会产生特殊的反射，在平滑石棉水泥墙特定角度上产生很不准确的距离测量。在机器人移动期间，这种相对角度以随机间隔发生。在装有多个声纳环的移动机器人中，尤其如此。在机器人运动时，一个声纳进入该误差模态的机会极高。从移动机器人的观点看，在这个情况下，声纳测量误差是随机误差；但是，如果机器人停止运动，则可能是完全不同误差模式。如果机器人的静态位置使一个特殊的声纳不能以一种方式工作，则声纳将一直地失误，并将一次又一次地精确地返回相同（和不正确）的读数。所以，一旦机器人静止不动，误差则表现为系统的，且是高精度的。

在这里，工作的基本机制是移动机器人对机器人姿态和机器人-环境动力学的交叉灵敏度。对这种交叉灵敏度的模型，其潜在意义不是真正随机的。然而，这种物理上的相互关系很少被建模，所以从非完整模型的观点看，在运动期间，误差表现为随机的；当机器人静止时，误差是系统性的。

声纳不是遭受系统性和随机误差模式模糊的唯一传感器。通过使用 CCD 摄像机的视觉理解，也高度地容易受机器人运动和位置的影响，因为摄像机依赖于光的变化、光的镜像性（比如，眩光）和反射。重要的一点是要认识到，系统误差和随机误差在受控的环境中是充分定义的，移动机器人可以呈现误差的特性，它沟通确定性和随机误差机制的差异。

多模误差分布 通常依据对不同输出值的概率分布来表征传感器随机误差的行为特性。一般，我们对随机误差的因果知之甚少，所

以普通使用几个简化的假设。例如，我们可以假定误差是零均的，即它对称地产生正和负的测量误差；我们可以进一步假定概率密度曲线是高斯型的。尽管我们在 4.2 节将详细讨论它的数学，现在重要的是认识人们经常假定**对称**和**单模分布**。这意味着测量正确值是最可能的，远离正确值的任何测量比接近正确值的任何测量可能性要小。这是硬性的假定，它使功能强大的数学原理能应用于移动机器人问题上。然而重要的是要了解这些假定通常是非常不适当的。

例如，我们再次考虑声纳传感器。当测量反射声音信号良好的物体的距离时，声纳会呈现高的准确度，并会产生基于噪声的随机误差，例如在定时电路中这部分的传感器行为将展现相当对称和单模的误差特性。然而，当声纳传感器通过环境而运动，且有时面对引起内在的反射而不是使声音信号返回到声纳传感器的那种材料时，则声纳会粗糙地过估计离物体的距离。在这种情况下，误差是非严格地系统性的，所以我们将此建模成随机误差的概率分布。因此，声纳传感器有两种不同类型的运行模式，第一种是信号确实返回且可能有某些随机误差；第二种是多路反射之后信号返回，且产生粗略的过估计。在这种情况下，至少概率分布容易地成为双模，且因为过估计比低估计更普通，它也是非对称的。

作为第二个例子，考虑用立体视觉的测距，我们再次可以认识到两个运行模式。如果立体视觉系统正确地将两个图像关联，那么最终的随机误差将由摄像机噪声产生，并限制测量准确度。但是，立体视觉系统也可能将两个图像**不正确地**关联起来，例如，匹配两个在现实环境中不相同的栅栏柱。在这种情况下，立体视觉会出现粗糙的测量误差，人们可以容易想象这个既破坏单模又破坏对称假设的行为。

本节的论点是，在移动机器人中，传感器**也许会**受到多模态的操作，因而在表征传感器误差时，很可能破坏单模性和对称性。但是，正如我们将看到的，许多成功的移动机器人系统，采用了这些简化的

假定和带有大量成功经验的实用数学技术。

上节已经提出了术语。用此，我们可以表征各种移动机器人传感器的优点和缺点。在下面一节，我们将对现在最常用的移动机器人的传感器的样品，也同样按此进行表征。

4.1.3 轮子/电机传感器

轮子/电机传感器是用于测量移动机器人内部状态和动力学的装置。这些传感器在移动机器人学之外有广泛的应用。因此，移动机器人享受了提供卓越分辨率、高质量、低价格的轮子和电机传感器所带来的好处。在下一节我们只提出一种传感器——光学增量编码器。

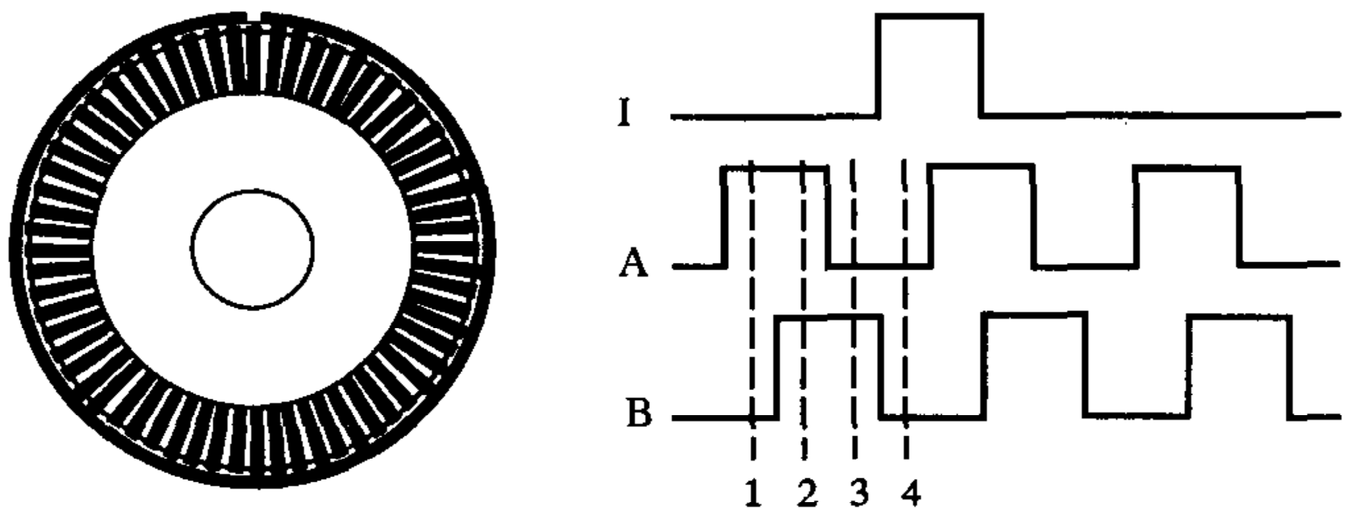
4.1.3.1 光学编码器

光学增量编码器已经成为在电机驱动内部、轮轴，或在操纵机构上测量角速度和位置的最普及的装置。在移动机器人学中，用编码器控制位置或轮子的速度，或其他电机驱动的关节。因为这些传感器是**本体感受式**的，在机器人参考框架中，它们的位置估计是最佳的。在用于机器人定位问题时，如在第5章讨论那样，需要重大的校正。

光学编码器基本上是一个机械的光振子。对各轴转动，它产生一定数量的正弦或方波脉冲。它由照明源、屏蔽光的固定光栅、与轴一起旋转带细光栅的转盘和固定的光检测器组成。当转盘转动时，根据固定的和运动的光栅的排列，穿透光检测器的光量发生变化。在机器人学中，最后得到的正弦波用阈值变换成离散的方波，在**亮**和**暗**的状态之间作选择。分辨率以**每转周期数**(CPR)度量。最小的角分辨率可以容易地从编码器的CPR额定值计算出。在移动机器人学中，典型的编码器可拥有2000 CPR，而光学编码器工业可容易地制造出具有10000 CPR的编码器。当然，根据所需的带宽，最关键的是编码器必须足够地快，以计算期望的轴转速。工业上的光编码

器不对移动机器人的应用提出带宽限制。

通常，在移动机器人学中，使用**正交编码器**。在这种情况下，第二对的照明源和检测器，按转盘被安放在相对于头一对位移 90°的地方。如图 4.2 所示，合成的一对方波，提供了有意义的更多信息。



状态	信道 A	信道 B
s_1	高	低
s_2	高	高
s_3	低	高
s_4	低	低

图 4.2 正交光学轮编码器：用所观测到的信道 A 和 B 脉冲链之间的相位关系确定旋转方向。外道的单一槽口产生每转的参考(指针)脉冲

按照哪个方波首先产生一个上升边沿进行排序，就可辨认转动方向。而且，四个可检测的不同状态，在不改变旋转盘情况下，分辨率可提高 4 倍。因此，一个 2000 CPR 正交编码器产生 8000 个计数。通过保留由光学检测器测得的正弦波并进行复杂的解释，有可能进一步改善分辨率。这种方法在移动机器人学中虽然少见，但分辨率却可以改善 1000 倍。

就大多数本体感受式传感器而言，编码器一般处在移动机器人内部结构受控的环境中，所以可以设计成无系统误差和无交叉灵敏度。光学编码器的准确度常常被认定为 100%，虽然这并不完全正确。但在光学编码器级别上，任何误差会因电机轴误差在下游而显得微不足道。

4.1.4 导向传感器

导向传感器可以是**本体感受式的**(陀螺仪、倾角罗盘)或外感受式的(罗盘)。它们被用来确定机器人的方向和倾斜度。与适当的速度信息结合在一起，容许我们把运动合并到位置估计。这个过程源

于船舶导航，被称为**航位推测法**。

4.1.4.1 罗盘

测量磁场方向的两个最普通的现代传感器是霍尔效应和磁通(量)闸门罗盘。如下所述，它们各有其优缺点。

霍尔效应描述了在出现磁场时半导体中电势的变化。当给横跨半导体的长度施加一个恒定电流时，根据半导体相对于磁通线的方向，则横跨半导体的宽度，在垂直方向就会有一个电压差。另外，电势的符号确定了磁场的方向。因此，单个半导体提供了一维磁通和方向的测量。在移动机器人学中，霍尔效应数字罗盘是很普遍的，且在直角方向包含两个这种半导体来提供磁场轴(起始端)的方向，从而获得八个可能之一的罗盘方向。该仪器不贵，但存在许多缺点。数字霍尔效应罗盘的分辨率很低，误差的内部源包括基本传感器的非线性和半导体电平系统性的偏移误差，最终的线路必须进行有效的滤波。这就把霍尔效应罗盘的带宽降低到对移动机器人而言是缓慢的值。例如，图4.3中所拍摄的霍尔效应罗盘，在90°转动后，需要经过2.5秒才稳定。

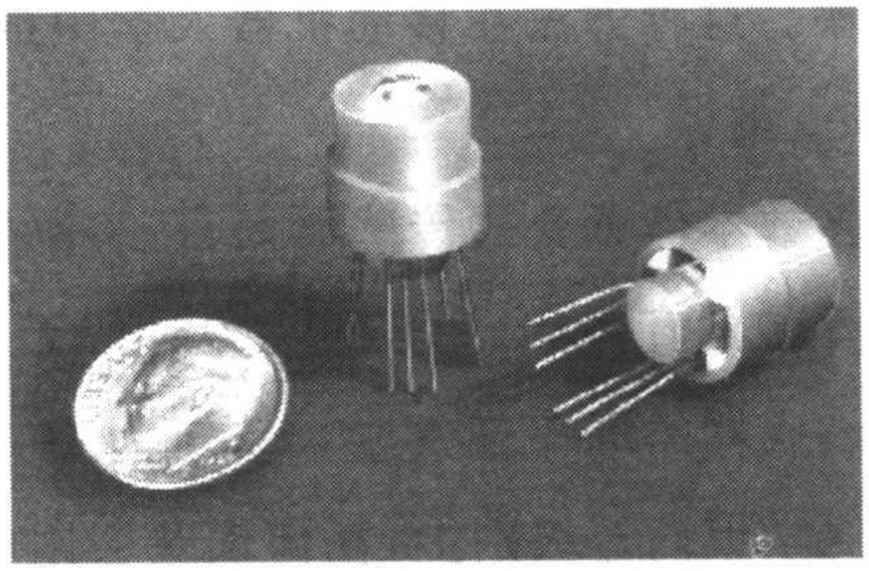


图4.3 数字罗盘：图中所示的为数字/模拟霍尔效应传感器，由Dinsmore公司提供(http://dinsmoregroup.com.dico)，能廉价(<15美元)产生感知磁场

磁通闸门罗盘根据不同的原理运行。两个小线圈绕在铁芯上相互垂直地装配。当交流电激励两个线圈时，根据各线圈的相对排列，磁场引起相移。测量相移，就可以计算二维的磁场方向。磁通闸门

罗盘可以准确地测量磁场强度，改善分辨率和准确度；但它比霍尔效应罗盘更大、更复杂。

不管所用罗盘的类型如何，移动机器人使用地球磁场时，其主要的缺点是涉及其他磁性物体和人造结构所产生的磁场干扰，以及电子罗盘带宽的限制和对振动的感受性。特别是在室内环境中，移动机器人的应用常常避免使用罗盘，虽然罗盘可以令人信服地提供有用的室内**局部的**方向信息，甚至出现钢结构时也是如此。

4.1.4.2 陀螺

陀螺仪是导向传感器，它可以用来保持相对于固定参考框架的方向。因此它们提供移动系统导向的绝对测量。陀螺仪可以被分成两类：机械陀螺仪和光学陀螺仪。

机械陀螺仪 机械陀螺仪的概念依赖于快速旋转转子的惯性性质。这种有趣的性质被称作陀螺仪的旋进。如果你试图让快速旋转的轮子围绕它垂直轴转动，你就会在水平轴上感觉到有刺激的反应。这是由与转轮相关的角矩引起的，并会保持陀螺轴惯性地稳定。反应力矩 τ，和随惯性框架的跟踪稳定性正比于旋转速度 ω、旋进速度 Ω 以及轮子的惯量 I。

$$\tau = I\omega\Omega \tag{4.5}$$

如图 4.4 所见，通过安装一个旋转轮，从外层轴枢到轮轴不能传输力矩。所以旋转轴是空间稳定（即被固定在惯性参考框架）的。然而，陀螺轴轴承的残余摩擦引入了小的力矩，因而限制了长期的空间稳定性，并在整个时间里产生小的误差。高质量的机械陀螺价格高至 100,000 美元，角度漂移大约为每 6 小时 0.1 度。

为了导航，必须一开始就选择旋转轴。如果旋转轴与南北子午线对准，那么地球的转动对陀螺仪的水平轴无影响。如果它指向东西，则水平轴会觉察到地球的转动。

速率陀螺仪具有图 4.4 所示的相同的基本结构，但稍有改动。万向节被一个具有附加阻尼的扭转弹簧所抑制，这使得传感器能测

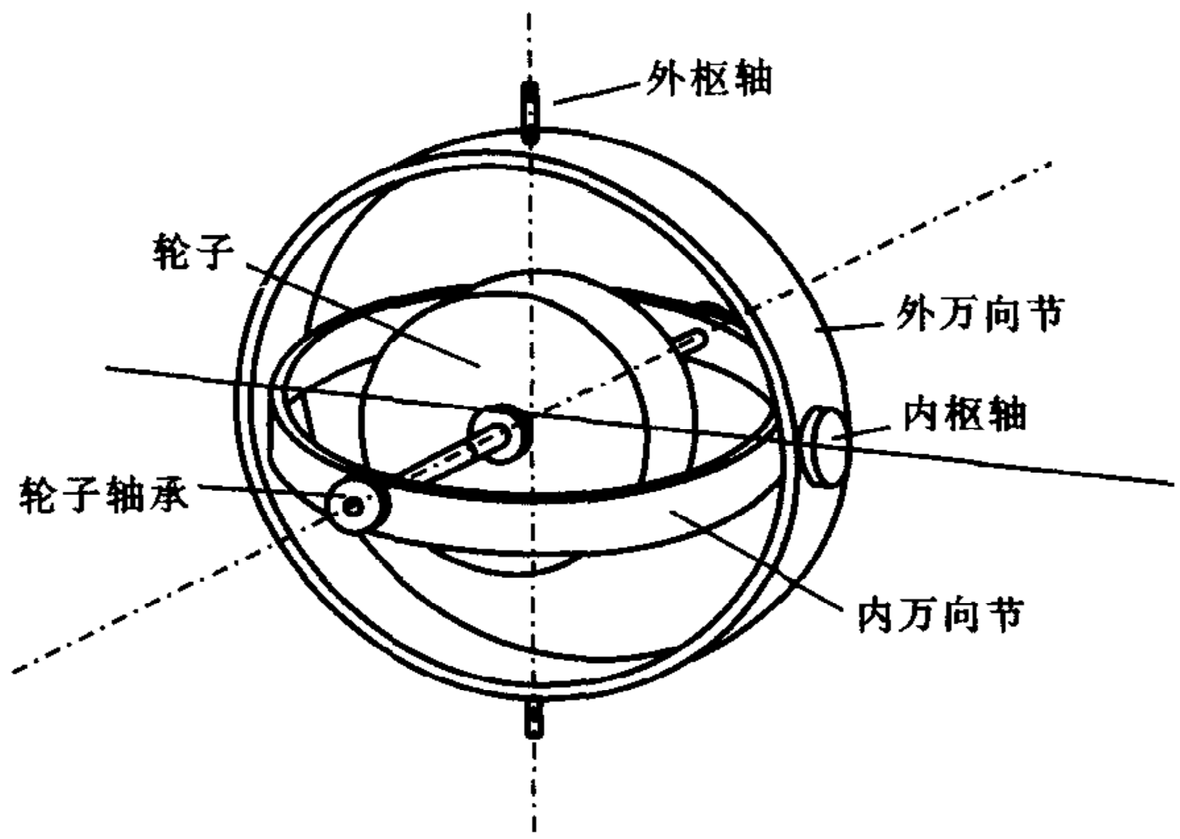


图 4.4 双轴机械陀螺仪

量角速度而不是绝对方向。

光学陀螺仪 光学陀螺仪是一个比较新的创造，于 20 世纪 80 年代初开始用于商业，那时它们首先被安装在飞机中。光学陀螺仪是角速度传感器，它们使用了从同一光源发射出来的两个单色光束或激光，而不是运动的机械部件。它们依据以下原理工作：光速保持不变，而几何特性的改变可以使光获得到达目的地的可变时间量。发射一个激光束，顺时针地行进通过一个光纤，而另一个逆时针地行进。因为行进在转动方向的激光路径稍短，所以它将有较高频率。两个光束的频率差 Δf 正比于圆柱体的角速度 Ω。基于相同原理的固体光学陀螺仪用微制作技术做成，所以提供了分辨率和带宽远远超过移动机器人应用所需要的导向信息。例如，带宽可以容易地大于 100 kHz，而分辨率可小于 0.0001°/h。

4.1.5 基于地面的信标

在移动机器人学中解决定位问题的一个很好的方法是使用有源或无源的信标。利用机载传感器和环境信标的交互，机器人可以精确地识别它的位置。虽然一般的直觉与以前人类导航的信标（如星座、山峰和灯塔）一样，但在规模为几公里的区域内，现代的技术已经

能使传感器定位室外的机器人，准确度优于 5 cm。

在下面这一节里，我们将描述一种信标系统，即全球定位系统(GPS)，它对室外的地面和飞行机器人极为有效。室内信标系统通常较少成功，其中有许多原因。在室内安置中，环境改造的费用对一个极大的有用区域，例如，像 GPS 情况一样是不合算的。而且，室内环境存在室外碰不到的重大困难，包括多路径和环境动力学。例如，一个基于激光的室内信标系统，必须从墙壁、平滑的地板和门反射出来的可能几十个其他强大的信号中，区分出一个真正的激光信号。令人困惑的是，人类或其他障碍物会经常不断地改变环境。例如，阻塞一条从信标到机器人的真正路径。在商用中，诸如在制造厂，需要慎重地控制环境以保证成功。在非结构的室内环境中，尽管已使用了信标，但问题仍要通过小心地布置信标和使用无源感知的模式予以缓解。

4.1.5.1 全球定位系统

全球定位系统(GPS)最初是为军事应用而开发，现在可免费地用于民用导航。它由总是存在的至少 24 个运行的 GPS 卫星组成。卫星每 12 小时在 20.190 km 高度沿轨道飞行。4 个卫星位于与地球赤道平面倾斜成 55°的 6 个各自平面中(图4.5)。

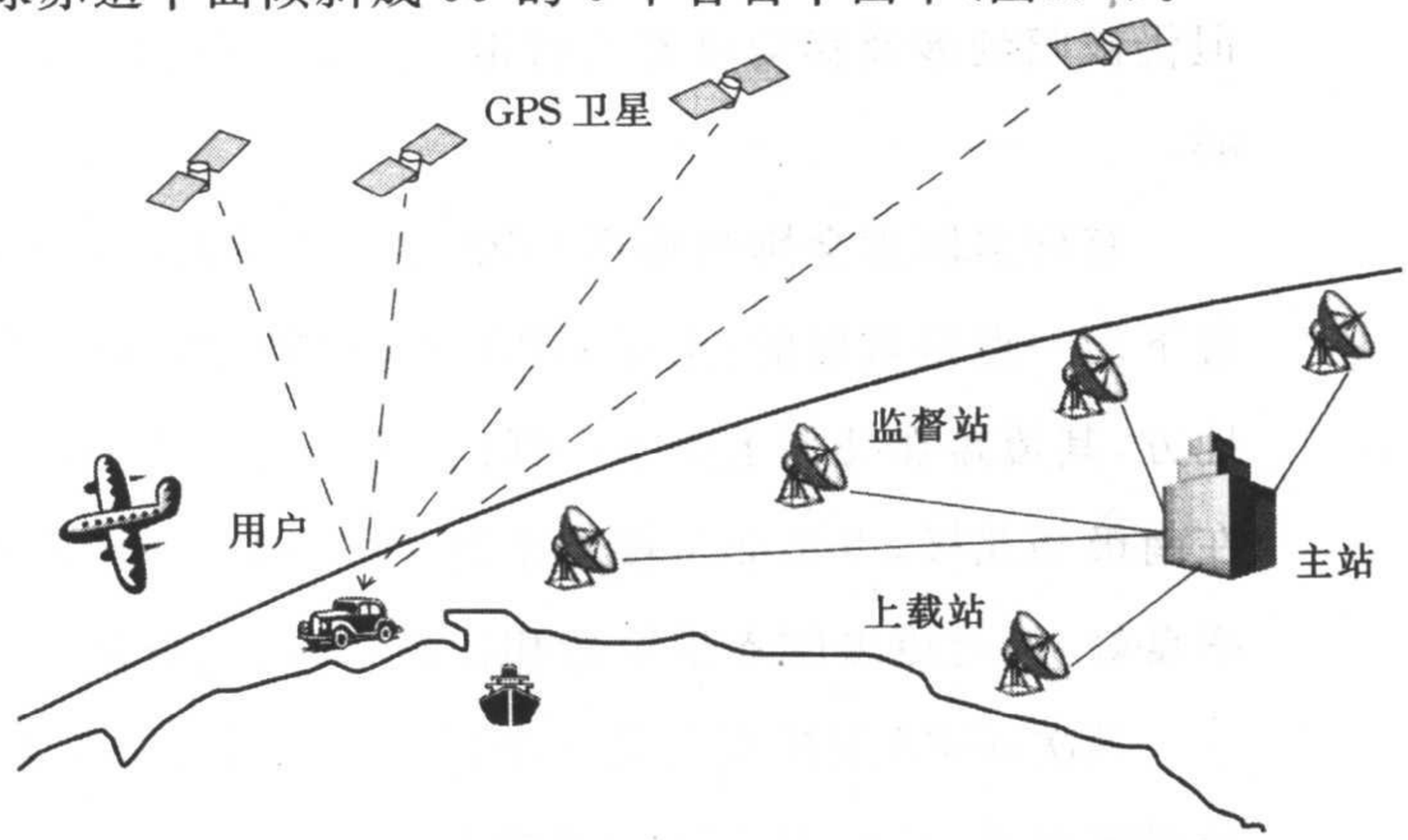


图 4.5 基于 GPS 的位置和方向的计算

各个卫星连续地发送指示其位置和当前时间的数据。所以GPS接收器是完全被动、但外感受式的传感器。当一个GPS接收器读取两个或两个以上卫星发送的数据时，到达的时间差作为各卫星的相对距离而告知接收器，组合关于到达时间和4个卫星瞬时位置的信息，接收器可推算出它自己的位置。理论上，这种三角测量只要求3个数据点。然而在GPS的应用中，定时是极端重要的，因为被测的时间间隔是纳秒。当然，卫星准确同步是强制的。为此，它们由地面站有规则地被更新，且各卫星都携带机载的定时原子钟。

GPS的接收器时钟也同样重要，它使我们能准确地测量各卫星发送的行进时间。但GPS的接收器只有一个简单的晶体时钟。所以，尽管3个卫星可理想地提供3轴位置，GPS接收器仍需要4个卫星，利用附加信息求解4个变量：3个位置轴和1个时间校正。

GPS接收器必须同时读取4个卫星传送的数据，这是一个很大的约束。GPS卫星传送功率极低，因而成功地读取数据要求与卫星直接进行瞄准(线)通信。在局限的空间里，诸如拥有高层建筑的市区或稠密的森林中，人们不大可能可靠地接收4个卫星。当然，大多数室内空间也不能为GPS接收器工作提供足够的天空可见度。由于这些原因，GPS虽然已成为移动机器人学中一个普通的传感器，但仍被归到涉及移动机器人行走于开阔空间和自主飞行器的那种范畴。

有许多因素会影响使用GPS定位传感器的特性指标。首先，理解下面一点是重要的：由于GPS卫星的特定轨道路径在地球不同的地方，其覆盖在几何上是不一样的，因而分辨率是不一致的。特别是在南极和北极，卫星非常接近于地平线，因而在经度和纬度方向分辨率良好，但与更多的赤道位置相比较，高度的分辨率比较差。

其次，GPS卫星仅仅是一个信息源。为了达到极其不同的定位分辨率水平，可以用不同的策略使用GPS。GPS使用的基本策略，称为**伪距离**。上面已述，一般分辨率达到15 m。该方法的扩展是差

分 GPS(DGPS),它使用了静态且位置精确已知的第二个接收器。利用这个参考接收器,可以校正许多误差,故分辨率提高到 1 m 数量级或更小。这个技术的一个缺点是必须安装一个平稳的接收器,它的位置必须仔细地予以测量。当然,为了从 DGPS 技术获得好处,移动机器人必须位于这个静态单元几公里范围内。

进一步改进的策略是考虑各接收卫星传送的载波信号的相位。在 19 cm 和 24 cm 有两个载波,所以,当成功地测量多个卫星之间相位差时,精度的大幅度改善是可能的。这种传感器对于点位置可以达到 1 cm 的分辨率;而用多个接收器,如在 DGPS 中,分辨率可达 1 cm以下。

移动机器人应用所考虑的最后一个因素是带宽。GPS 一般会有不小于 200～300 ms 的时延,所以可以期待有不高于 5 Hz 的 GPS 更新数据。在一个高速移动机器人或飞行机器人中,这可能意味着,由于 GPS 时延的限制,为了合适的控制,需要局部运动的积分。

4.1.6 有源测距

在移动机器人学中,有源测距传感器继续成为最流行的传感器。许多测距传感器价格低。最重要的是,所有测距传感器提供易于解释的输出:直接测量机器人到其邻区物体的距离。对于障碍检测和避障,大多数移动机器人严重依赖于有源测距的传感器。但是,由测距传感器提供的局部自由空间的信息,也可以累计到超越机器人当前局部参考框架之外的表示中。因此,也可以见到有源测距传感器一般地当作移动机器人定位和环境建模过程的一部分。但是,随着效果良好的视觉解释能力的缓慢出现,我们相信这类有源传感器在移动机器人专家选择传感器类别时,会逐渐失去其首要地位。

下面我们提出两种**飞越时间**的有源测距传感器:超声传感器和激光测距仪。然后提出两种几何有源测距传感器:光学三角测量传

感器和结构式光传感器。

4.1.6.1 飞越时间的有源测距

飞越时间测距利用了声或电磁波的传播速度。声音或电磁波的行进距离一般由下式给定：

$$d=ct \tag{4.6}$$

式中

d=行进的距离(通常往返旅行)；

c=波的传播速度；

t=飞越时间。

指出下面一点是重要的：声音的传播速度 v 近似为 0.3 m/ms，而电磁信号的速度是 0.3 m/ns，比声音快一百万倍。对于一个典型的距离，比如 3 m，超声系统的飞越时间是 10 ms，但对激光测距仪仅为 10 ns。显然，用电磁信号测量飞越时间，计算上更具挑战性。这就说明了为什么激光测距传感器只在最近对移动机器人的应用才变得可用和稳当。

飞越时间的测距传感器的质量主要取决于：

- 在确定反射信号精确到达时间中的不确定性；
- 在飞越时间测量中(特别是用激光测距传感器时)的不准确性；
- 发射束的传播圆锥体(主要对超声测距传感器)；
- 与目标的交互(例如，表面吸收和特殊反射)；
- 移动机器人和目标的速度(在动态目标的情况下)。

正如下面所讨论的那样，各类飞越时间的传感器对上列因素的特殊子集是敏感的。

超声传感器(声音飞越时间) 超声传感器的基本原理是发送(超声)压力波包，并测量该波包反射和回到接收器所占用的时间。引起反射的物体距离 d 可以根据声音传播速度 c 和飞越时间 t 进行计算，

$$d=\frac{ct}{2} \tag{4.7}$$

空气中声音的速度 c 由下式给定：

$$c=\sqrt{\gamma RT} \tag{4.8}$$

式中

γ=指定的热比率；

R=气体常数；

T=温度，单位为 K(绝对温度)。

在大气标准压力和 20℃条件下，声速近似为 $c=343$ m/s。

图 4.6 表示超声传感器不同的信号输出和输入。首先，发射一

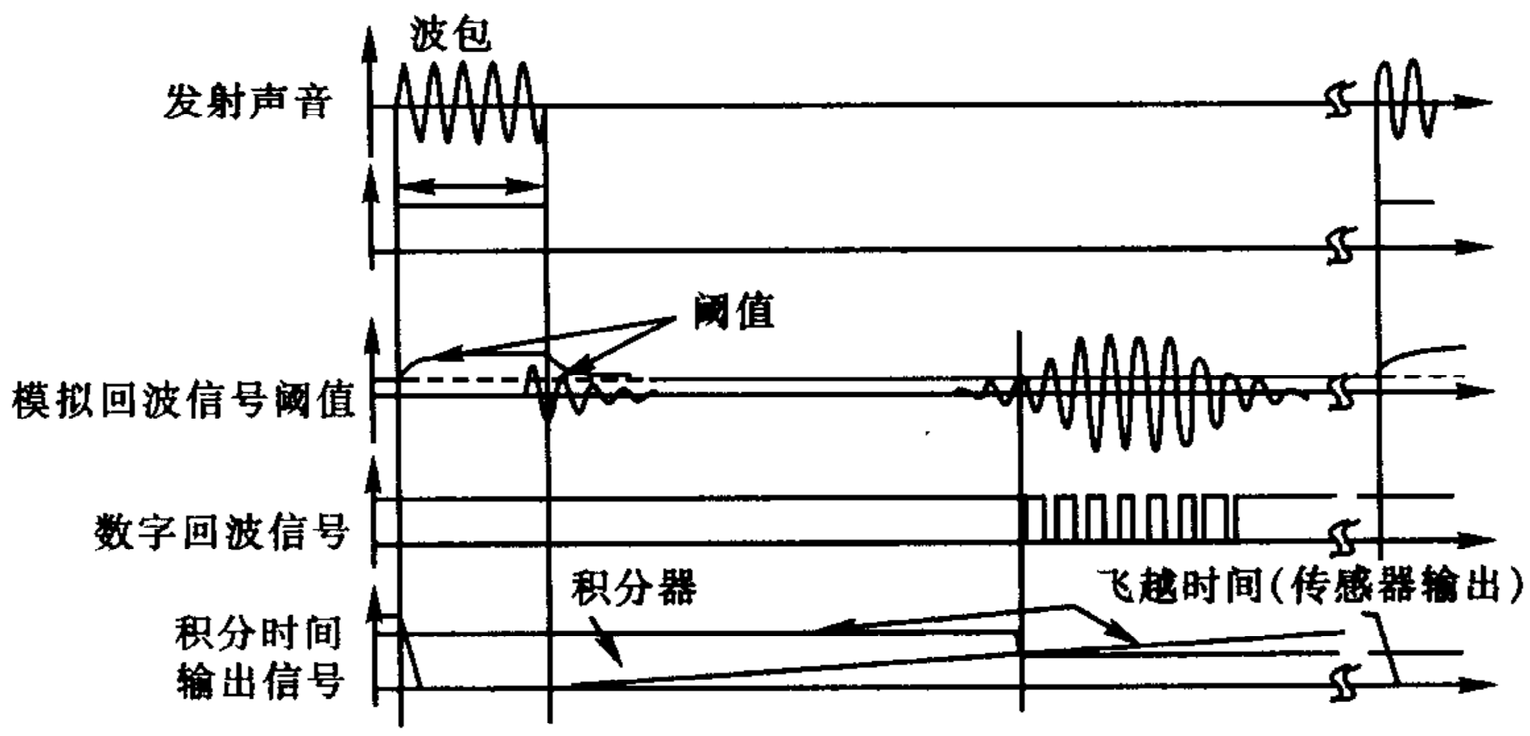


图 4.6 超声传感器的信号

系列声脉冲，组成**波包**。积分器的数值也开始线性地上升，测量这些声波从发射到回波检测的时间。为了把输入声波引发成有效的回波，设置了一个阈值。这个阈值通常随时间下降。因为根据分散过程，当回波行进较长距离时，期望的回波幅度随时间而下降。但在初始脉冲发送期间及发送短暂之后，将阈值设置到很高，以防止因向外发送的声脉冲触发回波检测器。变送器在初始发射之后继续循环，多至几个毫秒，从而控制传感器的**消隐时间**。注意，在消隐时间期间，如果发送的声音遇到距离极近的物体而反射，并返回到超声传感器，则检测可能失败。

然而，一旦消隐间隔通过，系统将检测任何高于阈值的反射声波，触发一个数字信号，并用积分器的值进行距离测量。

超声波典型地具有40～180 kHz之间的频率，通常由压电或静电变送器产生。虽然通过使用不同的输入和输出装置可以缩减所需的消隐间隔，但我们常常用相同的单元测量反射信号。当为移动机器人选择合适的超声传感器时，可以用频率来选择有用的范围。较低的频率相当于较长的距离，但有较长的后发送循环的缺点，因而需要较长的消隐间隔。移动机器人所用的大多数超声传感器有效距离大约为12 cm～5 m。商用超声传感器公布的准确度在98%～99.1%之间变化。在移动机器人应用中，用特定的实现方法，一般可达到近似为2 cm的分辨率。

在大多数情况下，为了同样得到关于所碰到物体的精确的方向信息，声束可能需要一个狭小的孔径角。这是一个主要的限制。因为声音以锥形方式(图4.7)传播，有大约20～40度的孔径角。所

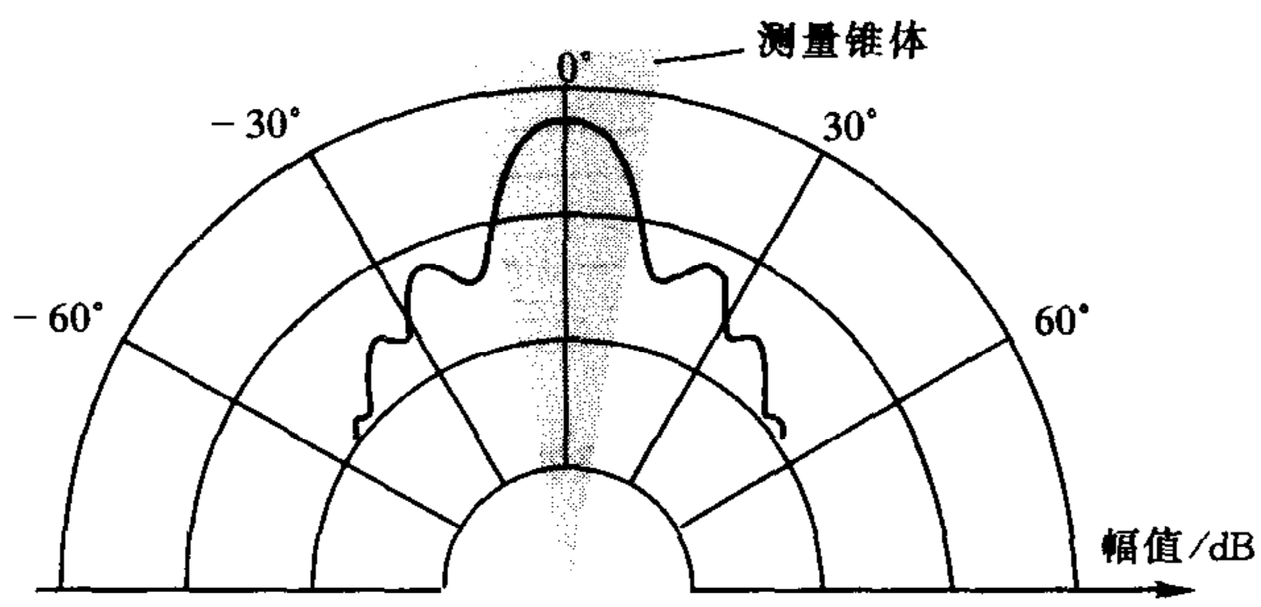


图4.7 超声传感器的典型强度分布

以，用超声测距时，人们没有获得深度的数据点，而是恒定深度的整个区域。这意味传感器只告诉我们在测量锥体内，在一定的距离，有一个物体。必须将传感器的数据画成圆弧(3D为球)的片断，而不是点测量(图4.8)。但最近研究发展表明，使用复杂回波处理[76]，测量的质量有重大改进。

超声传感器有另外几个缺点，即误差面积、带宽和交叉灵敏度。超声学公布的准确度，是基于离开声学上反射物质的声波是成功的、垂直的反射的名义值。这没有找准移动机器人经过环境运动所见到

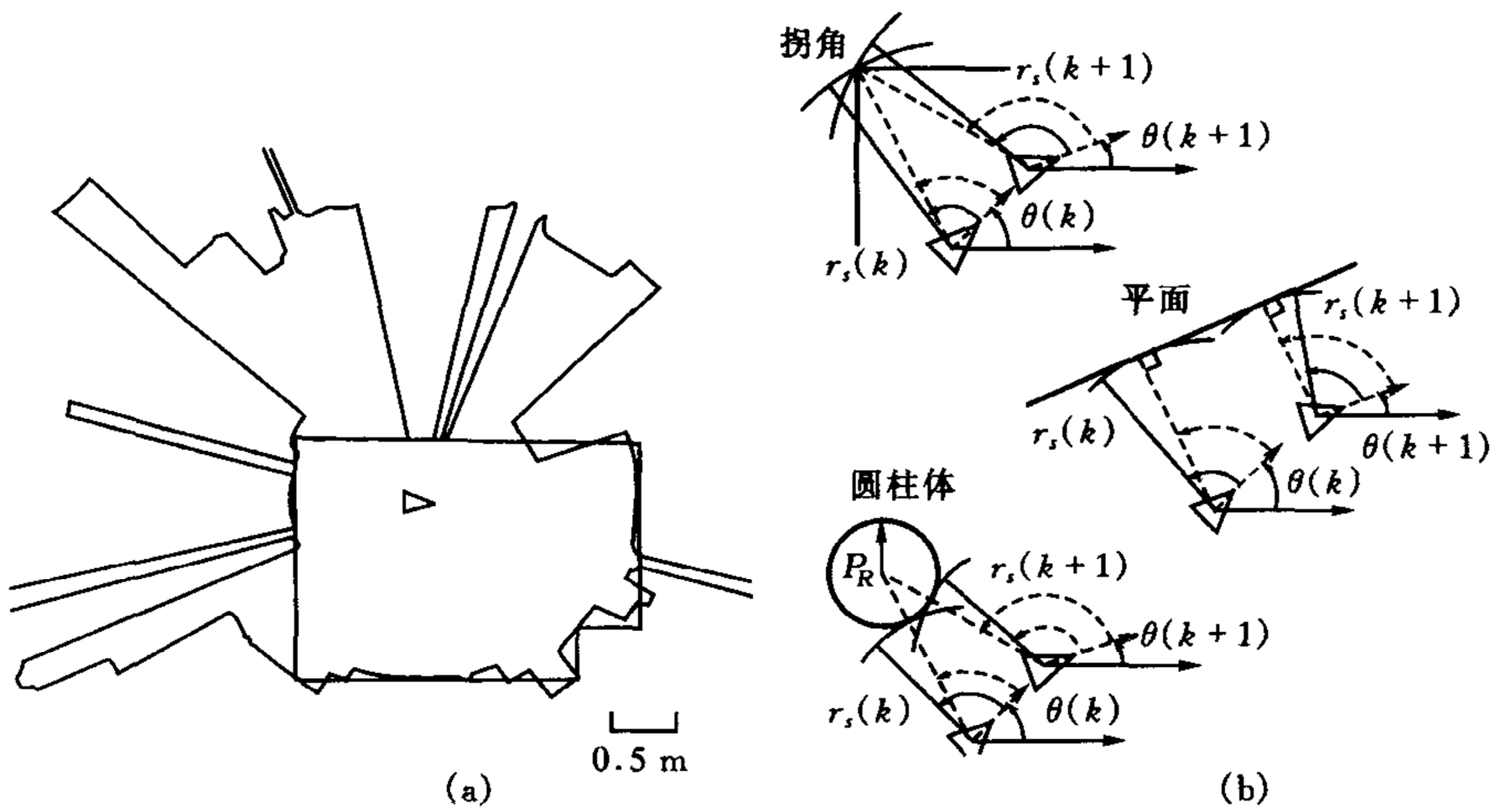


图 4.8 超声系统的典型读数:(a) 360°扫描;(b)不同几何原语得到的结果[23]。图片由 MIT 的 John Leonard 提供

的有效误差的特征。因为超声变送器对被测距物体的角度改变远非垂直,正像光线以浅角从光滑表面反射一样,声波离开传感器,相干反射的机会是良好的。在成功回射的情况下,如果充分了解接近真值的误差分布,就知道超声传感器的真正误差行为是复合的;而在相干反射的情况下,则更少了解距离值的集合,它比真值大很多。当然,被测距的物体材料其声学性质直接影响传感器的性能指标。再者,如果我们用一种材料,它可能不会产生单元感知很强的反射,则影响是离散的。例如,泡沫、毛皮和衣服,在不同的环境中,在声学上它们可以吸收声波。

超声测距的最后一个限制关系到带宽。特别是在中等开阔的空间中,单个超声传感器具有较慢的周期时间。例如,测量 3 m 以外物体的距离,这样的传感器需 20 ms,操作速度限制在 50 Hz。但是,如果机器人有一个具有 20 个超声传感器的环,各传感器顺序地激发和测量,使传感器之间的干涉最小,那么环的周期时间变为0.4 s,任何一个传感器总的更新频率只有 2.5 Hz。但用超声进行避障时,对一个引导中等速度运动的机器人,在依然可感知和安全避障的同时,这

样的更新速率对最大可能的速度会有一个可测量到的影响。

激光测距仪(飞越时间,电磁式) 激光测距仪是一个飞越时间传感器,由于使用激光而不是声音,相对于超声测距传感器,它得到了重大的改进。这种类型的传感器由发射器和接收器组成,前者用准直射束(即激光)照亮目标;后者能检测与发射光束本质上是同轴光的分量。这些装置通常被称作光学雷达或**激光雷达**(光检测和测距),它们可根据光到达目标然后返回所需的时间产生距离估计。一个带有镜子的机械机构扫掠光束,覆盖平面中所需场景,或者甚至在三维情况下,它使用一个旋转且摆动的镜子。

测量光束飞越时间的一种方法是使用脉冲激光,然后正像前述的超声解决方案一样,直接测量占用的时间。在这种器件中,需要能解决兆分之一秒的电子技术,所以它们非常昂贵。第二种方法是测量调频连续波(frequency-modulated continuous wave,FMCW)和它接收到的反射之间的差频。另一种更容易的方法是反射光的相移。下面我们详细描述第三种方法。

相移测量 近红外光(从光发射二极管(LED)或激光器发出)是从图 4.9 中的发射器校准和发射的,并点击环境中的 P 点。对粗糙

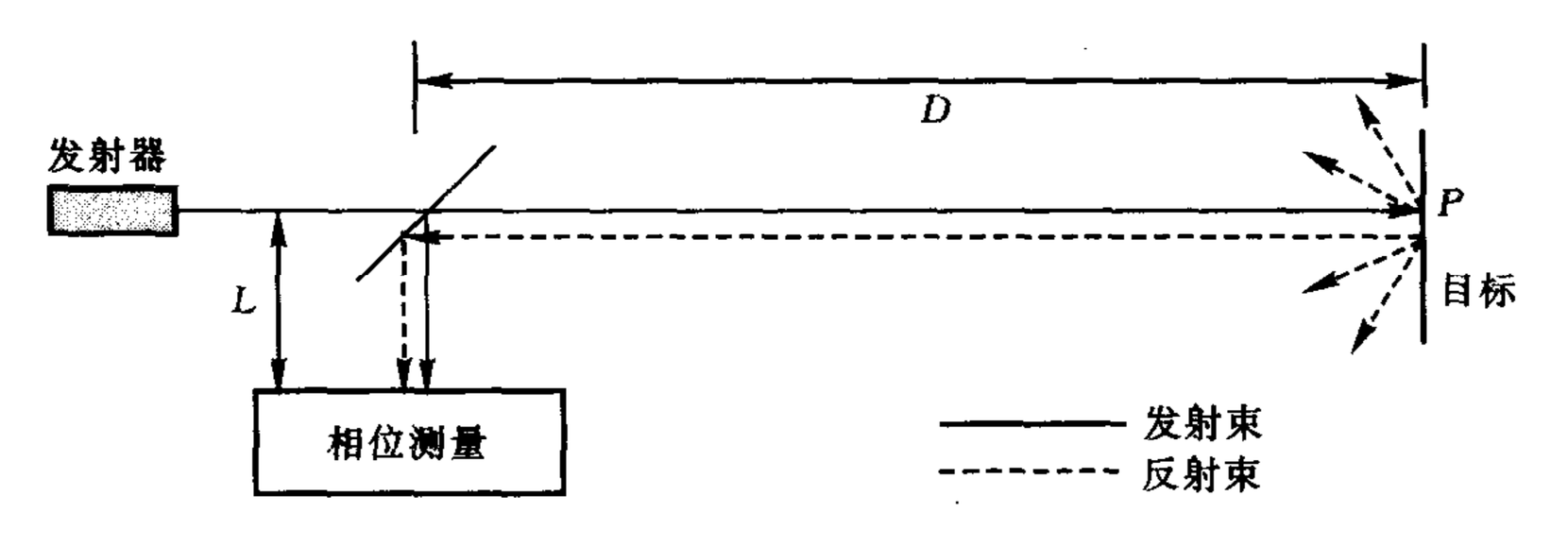


图 4.9 相移测量的激光测距仪的示意图

度大于入射光波长的表面,会产生漫射,这意味着光几乎各向同性地反射。发射的红外光波长是 824 nm。所以,除了只有高度抛光的反射物体外,大多数物体将是漫射反射器。落在传感器接收孔径的红外光分量,几乎是平行于远程物体所发射的光束而返回。

传感器以已知的频率发射 100%幅度的调制光，并测量发射和反射信号之间的相移。图 4.10 表示了该技术如何可用于测量距离。

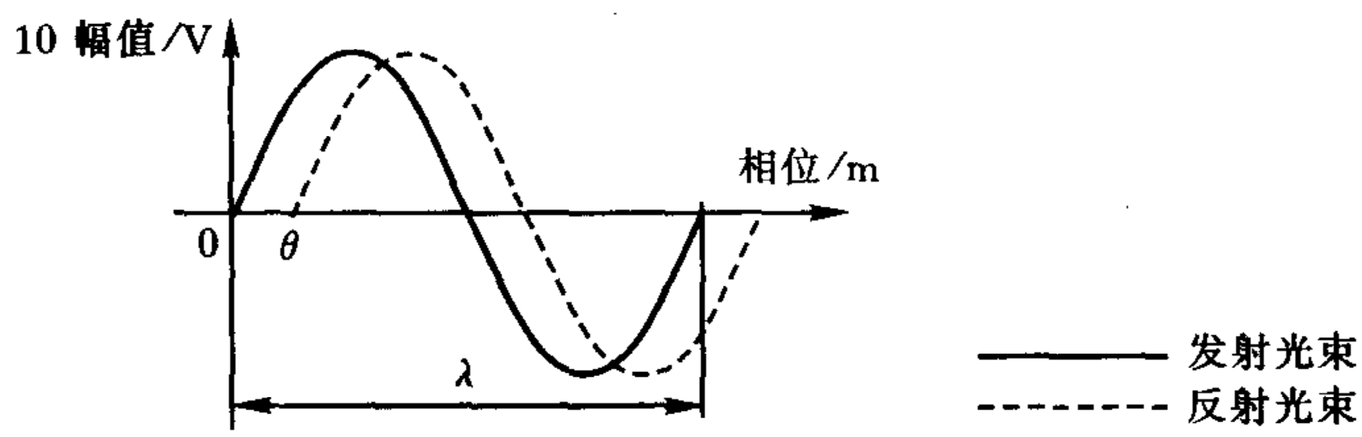


图 4.10 通过测量发送和接收信号之间的相移估计距离

调制信号的波长服从方程 $c=f\cdot\lambda$，式中 c 是光速，f 是调制频率。对 $f=5$ MHz(如 AT&T 的传感器)，$\lambda=60$ m，由发射光覆盖的总距离 D' 为

$$D'=L+2D=L+\frac{\theta}{2\pi}\lambda \tag{4.9}$$

式中 D 和 L 是图 4.9 所定义的距离。光束分离器和目标之间的所需距离 D 由下式给出：

$$D=\frac{\lambda}{4\pi}\theta \tag{4.10}$$

式中 θ 是电子测量的发射和反射光束间的相移差，λ 是已知的调制波长。可以看到，单个频率的调制波在理论上可以产生模糊的距离估计，例如，如果 $\lambda=60$ m，一个距离为 5 m 的目标，会给出与距离为 65 m 的目标一样的相移测量，因为各相角相隔 360°。所以我们要定义一个 λ 的“模糊间隔”。但是实际上我们注意到，传感器的距离，由于信号在大气中的衰减，比 λ 低得多。

可以证明，距离的置信度(相位估计)反比于所接收信号幅度的平方，直接影响传感器的准确度。因此，黑暗、远距离的物体不会产生与近距离、光亮物体那样好的距离估计。

在图 4.11 表示了典型的 360°激光距离传感器的示意图和两个实例。

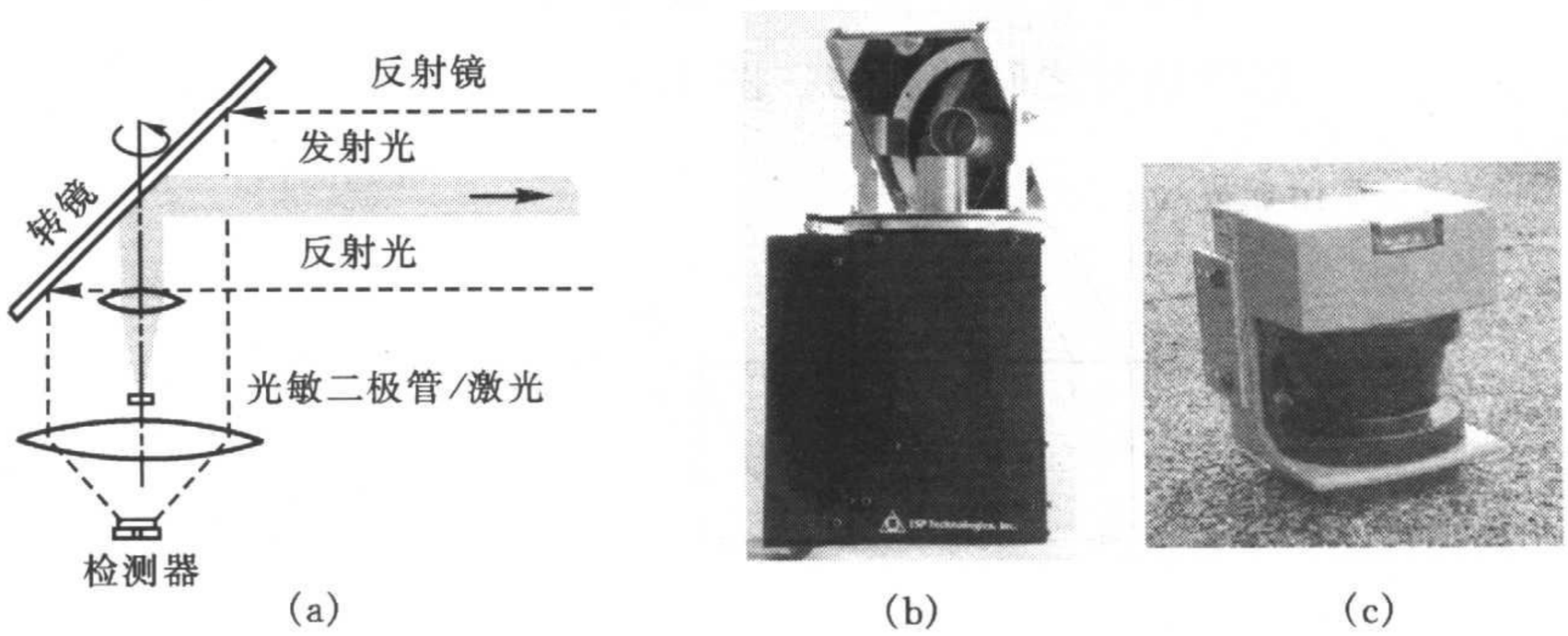


图 4.11 (a) 带有转动镜的激光距离传感器的示意图;(b) EPS 技术公司的扫描距离传感器;(c) 德国 Sick 公司的工业 180°激光距离传感器

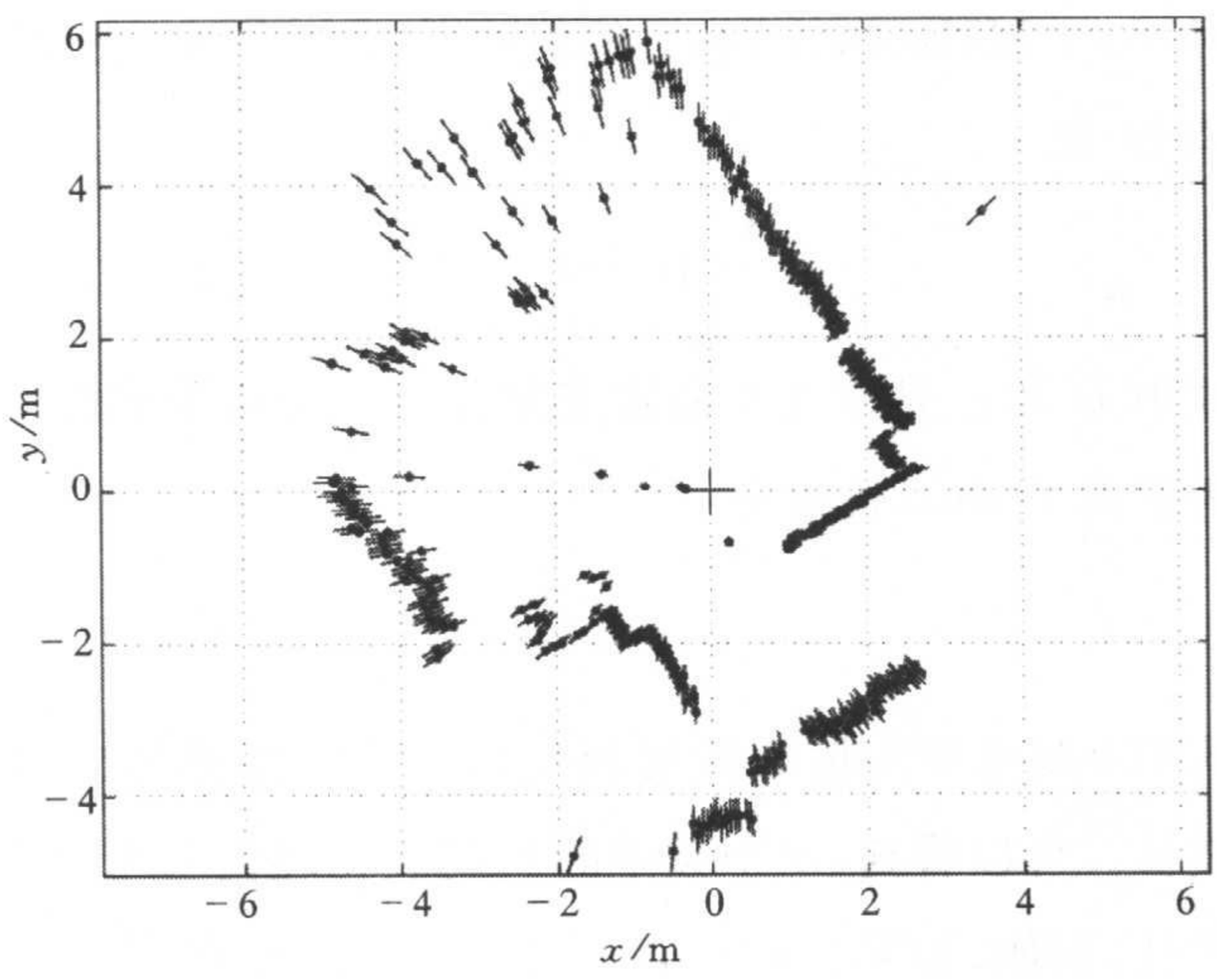


图 4.12 带有转动镜 2D 激光距离传感器的典型距离图像。经过测量点的直线长度说明不确定性

正如期待的那样,激光测距仪的角分辨率远远超过超声传感器。图 4.11 所示的 Sick 激光扫描仪角分辨率达到 0.5°。对于距离从 5 cm到 20 m 或更远,根据被测物体的亮度,深度分辨率近似为 5 cm。该装置每秒进行 25 个 180°的扫描,但对垂直的尺度没有镜子摆动的功能。

因为用超声测距传感器,所以一个重要误差模式涉及到能量的

相干反射；而用光，这只发生在撞击高度抛光表面的时候。实际上，移动机器人可能碰到以光泽桌面、文件柜或镜子那种形式的表面。与超声传感器不同，激光测距仪不能检测到光学上透明材料的出现，诸如玻璃。而这可能成为环境的一个重要障碍，例如博物馆，那里一般用玻璃。

4.1.6.2 基于三角测量的有源测距

基于三角测量的测距传感器利用它们在测量策略中显示的几何性质，对物体确定距离数据。基于三角测量的距离仪最简单的一类是**有源**的，因为它们把已知的光模式（例如，点、线或纹理）映射到环境。已知模式的反射被接收器捕获，并与已知几何值合在一起，系统可以利用简单的三角测量建立距离的测量。如果接收器沿着单个轴测量反射的位置，我们称该传感器为 1D 光学三角测量传感器；如果接收器沿着两个正交轴测量反射的位置，我们称此为结构式光传感器。在下面两节我们将描述这两个类型的传感器。

光学三角测量（1D 传感器） 如图 4.13 所示，光学三角测量的

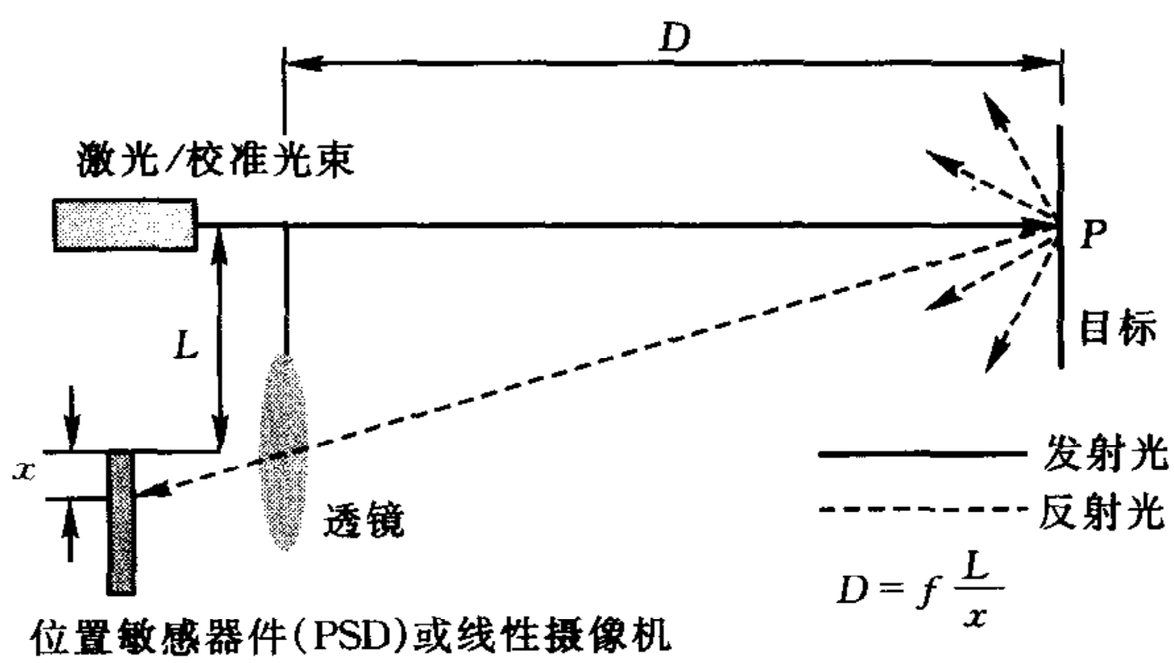


图 4.13 1D 激光三角测量的原理

原理是直截了当的。向目标发送一个已校准的光束（例如，聚焦的红外 LED，激光束）。反射光被透镜采集，并投影到一个位置敏感器件（position-sensitive device，PSD）或线性摄像机。给定图 4.13 的几何特征，则距离 D 给出如下：

$$D=f\frac{L}{x} \tag{4.11}$$

距离正比于$\frac{1}{x}$;所以传感器分辨率近距离最好,远距离就变差。使用基于这个原理的传感器,距离敏感高达1或2 m,但也可用于高精度的工业测量,分辨率远低于1 μm。

光学三角测量装置可以提供比较高的准确度,具有很好的分辨率(对近距离)。但是,这种装置的操作范围因几何特征而十分有限。例如,摄于图4.14的光学三角测量传感器操作在8～80 cm的距离范围。它比超声和激光测距传感器都便宜。虽然在距离上比声纳有更多的限制,但光学三角测量传感器具有高的带宽而且可以免受声域中普遍存在的交叉灵敏度的影响。

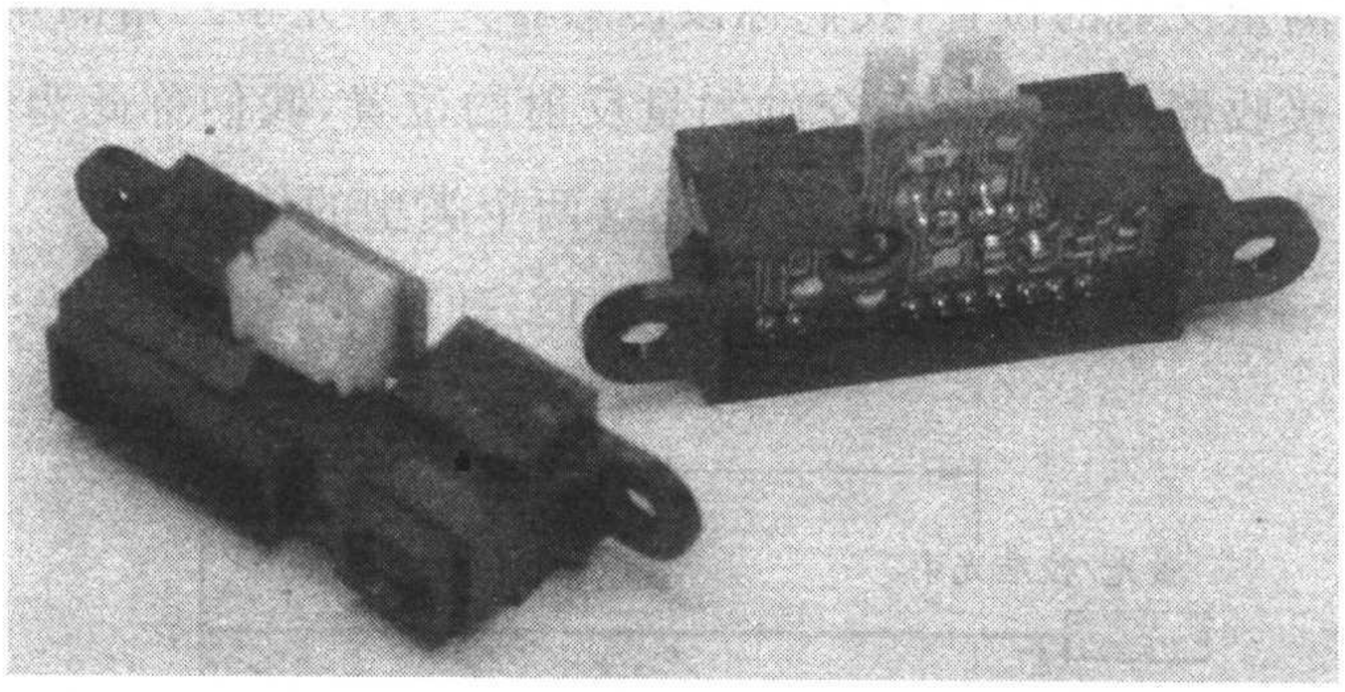


图4.14 商用、低价的光学三角测量传感器:Sharp GP系列红外测距仪提供模拟或数字距离测量,价格仅约15美元

结构光(2D传感器) 如果人们用2D接收器,诸如用CCD或CMOS摄像机替代线性摄像机或光学三角测量传感器的PSD,那么就可以对一大群的点而不只一个点测量距离。反射器必须将一个已知的模式或**结构光**投影到环境上。许多系统利用转动的镜子,或者投影光的纹理(图4.15(b))或者发射已校准的光(可能是激光)。还有另外一种选择,即用棱镜将激光束转变成一个平面,投影一个激光条纹(图4.15(a))。不管如何建立,被投影的光具有已知的结构,所以由CCD或CMOS接收器获得的图像可以被滤波以识别模式反射。

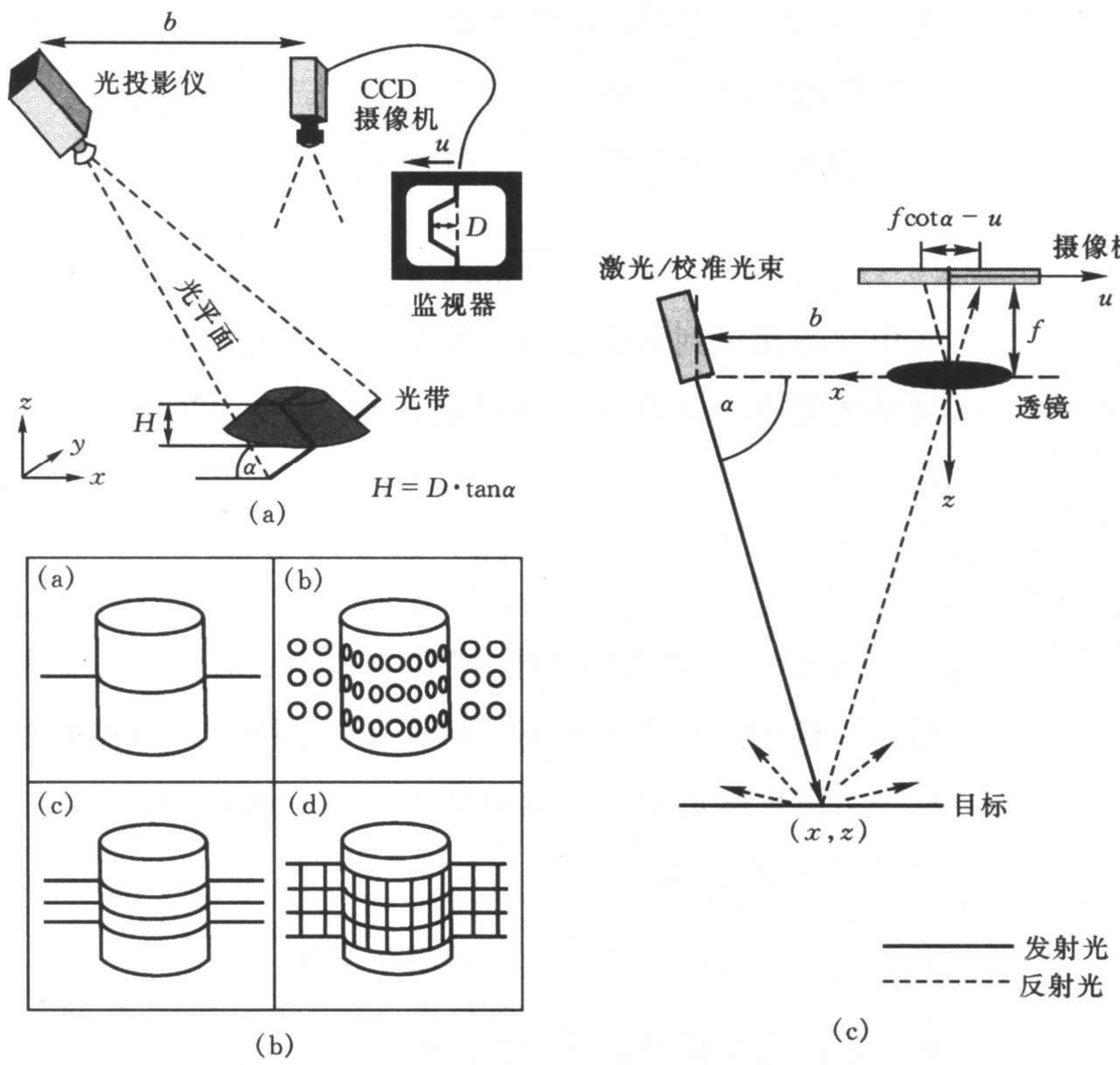


图 4.15 (a) 有源的二维三角测量的原理;(b) 其他可能的光结构;(c) 1D 的原理示意。图片由 Halmstad University 的 Albert-Jan Baerveldt 提供

注意到,在这个情况下,接收深度问题远比无源图像分析简单。在无源图像分析中,如以后我们讨论那样,必须用环境现有的特征进行校正,而现在的方法是将已知的模式投影到环境上,所以完全避免了标准的校正问题。而且,结构光传感器是一个有源装置,所以它会不间断地工作在黑暗环境以及物体无特征(例如,非均匀颜色和无边缘)的环境中。相反,立体视觉在无纹理的环境中会失误。

图 4.15(c)表示了一个 1D 的有源三角测量的几何特征。通过检验图 4.15(c)的几何特征,我们可以检查三角测量设计中的折衷。系统中被测量值是 α 和 u,u 是图像传感器中原点到被照明点的距

离。(注意,这里的图像传感器可以是摄像机或位置敏感器件的光子二极管的阵列。例如,一个 2D 的 PSD。)

从图 4.15(c),简单的几何特征表明,

$$x=\frac{b\cdot u}{f\cot\alpha-u};\ z=\frac{b\cdot f}{f\cot\alpha-u} \tag{4.12}$$

式中 f 是棱镜到图像平面的距离,在极限时,图像分辨率与距离分辨率之比被定义为三角测量增益 G_p,从方程(4.12),G_p 由下式给出:

$$\frac{\partial u}{\partial z}=G_p=\frac{b\cdot f}{z^2} \tag{4.13}$$

这表示,对给定的图像分辨率,测距准确度正比于源/检测器的间距 b 和焦距 f,并随距离 z 的平方而下降。在扫描测距系统中,存在一个由投影角 α 测量造成的对测距准确度的附加影响。从方程(4.12),我们知道,

$$\frac{\partial\alpha}{\partial z}=G_\alpha=\frac{b\sin\alpha^2}{z^2} \tag{4.14}$$

我们可以将参数对传感器准确度的影响归纳如下:

- **基线长度 b**:b 越小,传感器可越紧凑;b 越大,距离分辨率将越好。也要注意,虽然这些传感器没有遇到对应性问题,但不一致问题依然会发生。当基线长度 b 增加时,对近距物体,就会引入照明点也许不在接收器视场内的机会。

- **检测器长度和焦距 f**:较长的检测器长度可以提供或者较大的视场;或者改善的距离分辨率;或者二者的部分好处的结合。然而,增加检测器长度也意味着有较大的传感器头和较差的电气特性(增加随机误差和缩减带宽)。同样,短的焦距,以准确度为代价给出大的视场;反之亦然。

基于激光条纹的结构光传感器,在几个移动机器人的基地上曾经是通用的,因为相对于激光测距仪它是一个廉价的选择。随着在 20 世纪 90 年代激光测距仪质量的提高,结构光系统主要归到视觉

研究，而不是应用于移动机器人学。

4.1.7 运动/速度传感器

有些传感器直接测量机器人和它的环境之间的相对运动。因为这种运动传感器检测相对运动，只要物体相对于机器人的参考框架运动，运动就可被检测，它的速度就可被估计。有许多传感器，它们自然地测量运动或变化的某些方面。例如，热电传感器检测热的变化。当人步行经过传感器的视场时，他或她的运动，触发传感器参考框架中热的变化。在下一节中，我们描述基于多普勒效应的运动检测器的重要类型。这些传感器代表了支持它们的著名技术，具有几十年普遍应用的历史。对于快速运动的移动机器人，诸如自主的公路车辆和无人飞行器，基于多普勒的运动检测器就是障碍检测传感器的选择。

4.1.7.1 基于多普勒效应的感知(雷达或声音)

任何人，如果已经注意到当正在到来的消防车经过和后退时所发出的报警音调的变化，则就熟悉了多普勒效应。

一个发射器以频率 f_t 发射电磁波或声波。它或者被接收器(图 4.16(a))接收；或从物体(图 4.16(b))反射。在接收器，按照式(4.15)，测量的频率 f_r 是发射器和接收器之间相对速度 v 的函数。如果发射器正在运动，则

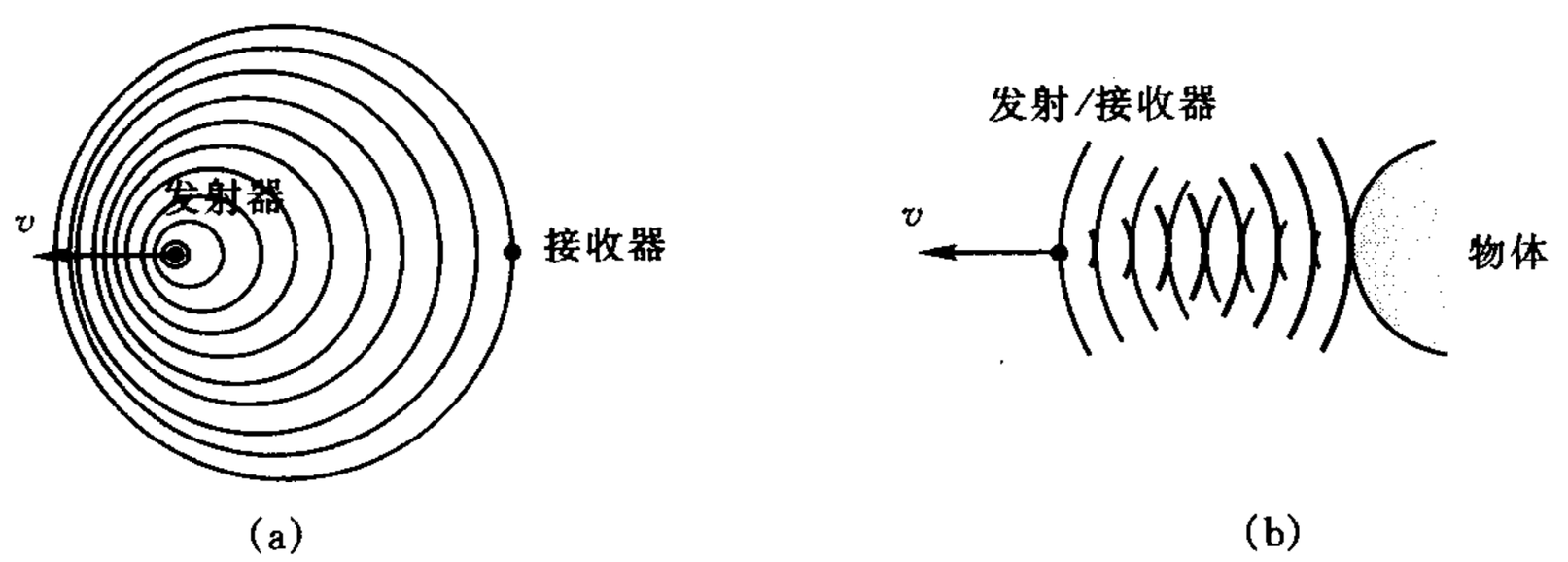


图 4.16 (a) 两个运动物体之间的多普勒效应；(b) 运动和静止物体之间的多普勒效应

$$f_r = f_t \frac{1}{1+v/c} \tag{4.15}$$

如果接收器正在运动，则

$$f_r = f_t(1+v/c) \tag{4.16}$$

在反射波的情况下(图 4.16(b))，引入了因子 2，因为在相对的间距中，x 的任何改变影响往返的路径长度为 $2x$。而且，在这种情形下，通常更方便考虑频率 Δf 的变化，相对于上面**多普勒频率**概念，这被称为**多普勒偏移**。

$$\Delta f = f_t - f_r = \frac{2 f_t v \cos\theta}{c} \tag{4.17}$$

$$v = \frac{\Delta f c}{2 f_t \cos\theta} \tag{4.18}$$

式中

Δf=多普勒频率偏移；

θ=运动方向和光束轴之间的相对角度。

多普勒效应使用声波和电磁波，它有广泛的应用范围：

- **声波**：例如，工业过程控制，安全，寻鱼，测量地速。
- **电磁波**：例如，振动测量，雷达系统，对象跟踪。

现在的应用领域是自主的和有人的公路车辆两个方面。对这类环境，已经设计了微波和激光雷达的两种系统。这两种系统具有等效的量程，但当视觉信号被环境条件，如下雨、雾等恶化时，激光却可以承受。商业上的微波雷达系统已被安装到公路卡车上。这些系统被称为 VORAD(车载雷达)，总距离近似为 150 m，准确度约 97%。这些系统报告，以 0～160 km/h 的测距速率，分辨率为 1 km/h。光束近似为 4 度宽，5 度高。雷达技术的主要限制是它的带宽。现有系统可以近似地以 2 Hz 提供多目标的信息。

4.1.8 基于视觉的传感器

视觉是我们最强的感知，它给我们提供数量巨大的关于环境的

信息。在动态环境中，能进行丰富的和智能的交互。所以不用惊奇，大量的力量都致力于制造一种机器，它具有模拟人类视觉系统的传感器。这过程的第一步是创造感知器件，它获取人类视觉系统所用的相同的原始信息光。下节我们描述两种制作视觉传感器的最新技术：CCD 和 CMOS。当与人眼相比较时，这些传感器在特性指标上有特定的限制。此后，第 2 节和第 3 节描述商业上已可用的基于视觉的传感器。像在本章以前所讨论的传感器一样，也会指出它们的缺点和最普通的应用。

4.1.8.1 CCD 和 CMOS 传感器

CCD 技术 电荷耦合器件是今天机器人视觉系统最流行的基本成分。CCD 芯片(图 4.17)是一个光敏像素元或像素的阵列，通常

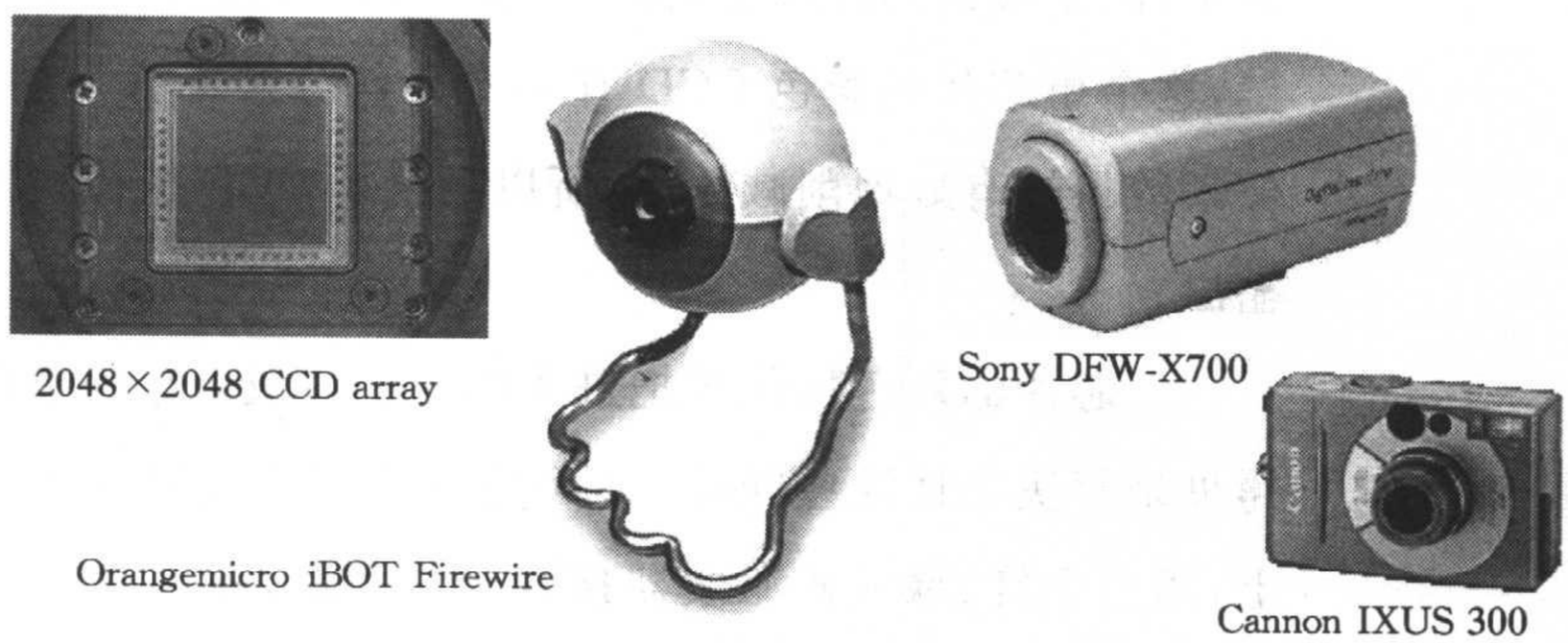


图 4.17 商业上已可用的 CCD 芯片和 CCD 摄像机。因为该技术比较成熟，所以摄像机具有变化繁多的形式和价格。(http://www.howstuffworks.com/digital-camera2.htm)

总数在 20,000 和几百万像素之间。各个像素可以被想象为是一个光敏不充电的电容器，尺寸为 5～25 μm。开始，所有像素的电容器全部充电，然后积分周期开始，当光的光子撞击各像素时，它们释放电子，电子被电场捕获并保留在像素上。随着时间的过去，根据撞击像素的光子总数，各像素累计电荷变化电位。在积分周期完成之后，所有像素的相对电荷需要被冻结和被读取。在 CCD 中，读取过程在 CCD 芯片的一角进行，像素电荷的底部行被传送到该角落并被读

取，然后，上面的行移下来，重复此过程。这意味着各电荷必须越过芯片而被传送，保持数值是至关重要的。这需要特别的控制电路和专门的制作工艺以确保被传送电荷的稳定性。

用在 CCD(以及 CMOS)芯片上的光敏二极管对所有光频不是同等敏感的。它们对 400～1000 nm 波长之间的光敏感。重要的是记住，光敏二极管对光谱的紫外端(比如蓝色)敏感低；而对红外部分(比如热)极为敏感。

你可以知道，基本的光测过程是无色的；这只是在积分周期，测量撞击各像素的光子总数。有两个通用的建立**彩色**图像的方法。如果在 CCD 芯片上的像素分组成 4 个 2×2 集合，那么红、绿和蓝染色可以被应用于彩色滤波器，使得各单独像素只接收一种颜色的光。通常，两个像素测量绿色，而一个像素各测量红色和蓝色的光强。当然，这个单芯片的彩色 CCD 有一个几何分辨率方面的缺点，系统中像素数目被有效地削减 4 倍，所以牺牲了 CCD 摄像机图像分辨率的输出。

三芯片的彩色摄像机把进入的光分裂成 3 个完整(较低强度)的拷贝来解决上述这些问题。3 个独立的 CCD 芯片，在各完整的芯片上，用一个红、绿或蓝滤波器接收光。因此，各芯片平行地测量一种颜色的光强，因而摄像机必须组合 CCD 芯片的输出以建立联合的彩色图像。在该解决方案中，保持了分辨率。正如大家所预料到的，三芯片彩色摄像机更昂贵，因此在移动机器人学中更少应用。

不论是三芯片还是单芯片的彩色摄像机，都有以下的缺陷：光敏二极管对光谱中近红外端的光更为敏感，这意味着整个系统检测蓝光比红光和绿光要差得多。为了补偿，在蓝色通道必须增强增益，这就在蓝色通道中引入了比红色和绿色更大的绝对噪声。普遍认为在蓝色通道至少有 1～2 位的附加噪声。虽然今天对这问题没有满意的解决方案，但随着时间的积累，增长蓝色的检测过程已有所改善，我们期待这个积极的趋向能继续下去。

CCD摄像机有几个影响其行为特征的摄像机参数。在某些摄像机中，这些参数是固定的；在另外一些中，根据内在的反馈回路其值是不断改变的。在更高档的摄像机中，用户可通过软件修改这些参数的值。**光圈位置和快门速度**调整被摄像机所测的光的总量，光圈只是一个机械上的孔径，像标准的35 mm照相机一样，它限定进入的光量。快门的速度调节芯片的积分周期。在高档的摄像机中，有效的快门速度可以短至1/30,000 s，长至2 s。**摄像机的增益**，在A/D之前，控制模拟信号总的放大倍数。然而，非常重要的是要了解，设置高增益之后，虽然图像也许显得更亮，但快门速度和光圈也许根本不改变。因此，增益仅仅放大信号。随着信号的放大，也放大了所有有关的噪声和误差。虽然在人们消费的摄像应用场合，增益是有用的(例如，照相、电视)，但对移动机器人专家来说，增益的价值不大。

在彩色摄像机里，有一个**白平衡**的附加控制。根据场景的照明源(例如，荧光灯、白炽灯、阳光、水下过滤光等等)，定义纯白色光的红、绿和蓝光的相对测量会极大地变化。人的眼睛补偿所有这些效应的原理还没有被充分了解。但是，摄像机可以展示光泽的不一致性。同一个桌子，在晚上照相时，在一张相片上看是蓝色；在白天照时，在另一张相片上看是黄色的。为了在变化的背景中，保持更一致的颜色定义，白平衡控制能使用户改变红、绿和蓝的相对增益。

CCD摄像机的主要缺点在于不一致的区域和动态范围。如上所述，许多参数可以改变摄像机创作图像的亮度和颜色。有保留地操作这些参数，可以对时间和环境提供一致性。例如，保证一件绿色衬衫看上去一直是绿的；黑灰色的东西总是黑灰的。这在视觉界留下了一个开放性的问题。关于颜色一致性和照明一致性的详细内容请参考文献[40]。

第二类缺点关系到在极端照明环境中CCD芯片的行为特征。在照明很低的情况下，各像素只接收少量数目的光子。最长可能的

积分周期(即快门速度)和摄像机的光学特性(即:像素大小、芯片尺寸、镜头的焦距和直径)将决定在信号强于随机误差噪声时,光的最低水平。在非常强的照明情况下,像素将使它的陷阱充满自由电子,并当陷阱达到它的极限时,诱捕附加电子的概率降低,所以陷阱中输入光和电子间的线性度退化,这叫做**饱和**。它可以说明有关交叉敏感度更深入的问题。当一个陷阱已达到其极限时,在积分周期的残余部分内,附加的光会造成更多的电荷漏到相邻的像素中,使它们产生不正确的值,或者甚至达到二次饱和。这个效应叫做**爆增**,意味着单独的像素值不是真正独立的。

针对具有特别亮度的环境,可以调整摄像机的参数,但摄像机动态范围受单个像素陷阱容量限制的问题仍然存在。例如,一个高质量的 CCD 会有持有 40,000 电子的像素,对数据的噪声水平,陷阱可以是 11 个电子。所以,动态范围将是 40,000 : 11,或 3600 : 1,即 35 dB。

CMOS 工艺技术 互补金属氧化半导体芯片与 CCD 有重大区别。它也有像素阵列,但是位于各像素旁的是像素特定的几个晶体管。正像 CCD 芯片一样,在积分周期期间,所有像素累积电荷。在数据收集步骤中,CMOS 采用新的方法:每个像素旁边的像素专用电路测量和放大像素的信号,且它对阵列的每个像素都是平行地进行的。利用从一般半导体芯片来的比较传统的步骤,将最终的像素值带到它们的目的地。

CMOS 具有超过 CCD 技术的几个优点。第一且是首要的,它不需要 CCD 所要求的特殊时钟驱动器和电路,要将各像素的电荷顺着阵列所有的列进行转移,并遍历所有行。这也意味着,CMOS 不需要专门的半导体生产过程来制作芯片。所有制作微型芯片的相同生产线同样可以制造廉价的 CMOS(图 4.18)。CMOS 芯片是如此简单,以至它消耗非常少的功率。令人难以置信的是,它运行的功耗,是 CCD 芯片功耗的百分之一。在移动机器人中,功率是稀有的

资源，因此，这是一个重要的优点。

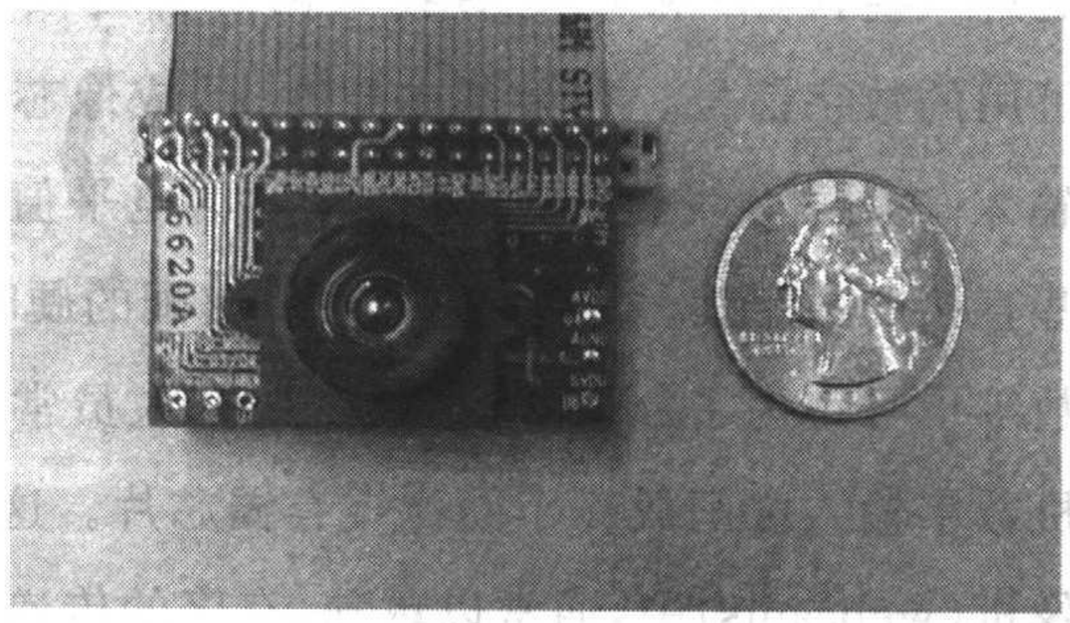


图 4.18 商业上已有的，附带镜头的低价 CMOS

另一方面，CMOS 芯片也面临几个缺点。最重要的是按照光检测阵列的字面意义上看，各像素旁边的电路消耗了有价值的实际资源。许多光子撞击晶体管而不是光敏二极管，使得 CMOS 芯片比等价的 CCD 芯片更灵敏。其次，CMOS 工艺技术比较年轻。因此，以 CMOS 格式可以获得的最佳分辨率仍然远比现有的最佳 CCD 芯片差得多。时间无疑将会促使高档 CMOS 的成像器更接近于 CCD 成像的特性指标。

给出了支持 CCD 和 CMOS 芯片机制的上述概要，我们可正确地评价任何基于视觉的机器人的传感器对其环境的灵敏度。与人眼相比，这些芯片的适应性、交叉灵敏度和动态范围全要差得多。结果，今天的视觉传感器仍然脆弱。只有随着时间的推移，当成像芯片的基本特性指标改善时，才会有移动机器人可用的、鲁棒性更好的基于视觉的传感器。

摄像机输出考虑 虽然数字摄像机本质上具有数字输出，整个 20 世纪 80 年代以及 90 年代初，大多数可提供的视觉模块却产生模拟输出信号，诸如 NTSC（全国电视标准委员会）和 PAL（逐行倒相）。这些摄像机系统包含一个 D/A 转换器，具有讽刺意义的是，它用帧接收器削弱计算机的作用，实际上它将 A/D 转换器板安置在计算机的总线上。D/A 和 A/D 远不是无噪声的，而且在这种摄像机

里，模拟信号的颜色深度是对人的视觉而不是对计算机的视觉优化。

更近一些，CCD和CMOS技术两者都提供可以被机器人专家直接可用的数字信号。在最基本层次上，成像芯片提供平行的数字I/O(输入/输出)引线，它传送离散像素的电平值。某些视觉模块利用这些直接的数字信号，但受成像芯片支配的硬时间约束，必须处理这些数字信号。为了放松实时要求，研究人员常常在成像器输出和计算机数字输入之间放一个**图像缓存器芯片**。这种芯片普遍用在网络摄像机(webcams)中，通常以单独、有序的传递方式捕获全图像快照，并能非实时地存取像素。

在最高层次，机器人专家可以作另一种选择，他们利用高级数字传输协议与成像器进行通信。虽然某些较老的成像模块也支持串行口(RS-232)，但是最普通的是IEEE 1394(5线)标准和USB(和USB 2.0)标准。为使用任何这种高级协议，人们必须为通信层和成像芯片特殊的实现细节确定或创造驱动码。然而，要注意到无损数字视频和为人类视觉消费而设计的标准数字视频流之间的区别。大多数数字视频摄像机提供数字输出，但常常只是压缩形式。对视觉研究人员而言，必须避免这种压缩，因为这不仅丢失了信息，而且甚至引入实际不存在的图像细节，诸如MPEG(Moving Picture Experts Group，运动图像专家组)的离散化边界。

4.1.8.2 视觉测距传感器

在移动机器人学中，距离感知是极其重要的，因为这是成功避障的基本输入。如同本章前面我们所看到的那样，许多传感器，如超声、激光测距仪、光学测距仪等等，之所以普遍地用于机器人学中，显然是因为它们具有获取深度估计的能力。自然，人们也力图用视觉芯片实现测距的功能。

但是，视觉图像的基本问题使得测距比较困难。任何视觉芯片把3D的环境压缩成一个2D图像平面，就失去了深度信息。如果有人能做出关于环境中物体的尺寸或它们的颜色和反射性的强假设，

那么人们就可直接解释 2D 图像，获取深度。但是，这种假设在现实环境的机器人应用中几乎不可能。没有这种假设，单独的照片无法提供足够的信息以恢复空间信息。

一般的解决方案是通过观察情景的几张图像以取得更多的信息，它足以有希望地，至少部分地恢复深度。所用的图像必须不一样，以使得合在一起时它们能提供附加信息。它们可以从不同的观察点获得**立体或运动**算法。另一种选择是建立不同的图像，但不改变观察点，而是改变摄像机的几何特性，诸如焦点位置或镜头的光圈。这是支持景深和散焦深度的基本概念。

在下节，我们略述景深技术的一般方法，因为它提供了一个直接有效的方法来制作基于视觉距离传感器。接着，我们从立体和运动入手，提出基于对应性的深度技术的细节。

景深 景深技术的分类依赖于以下事实：图像性质不仅作为情景的函数而改变，而且也是摄像机参数的函数。摄像机参数和图像参数之间的关系表示于图4.19。

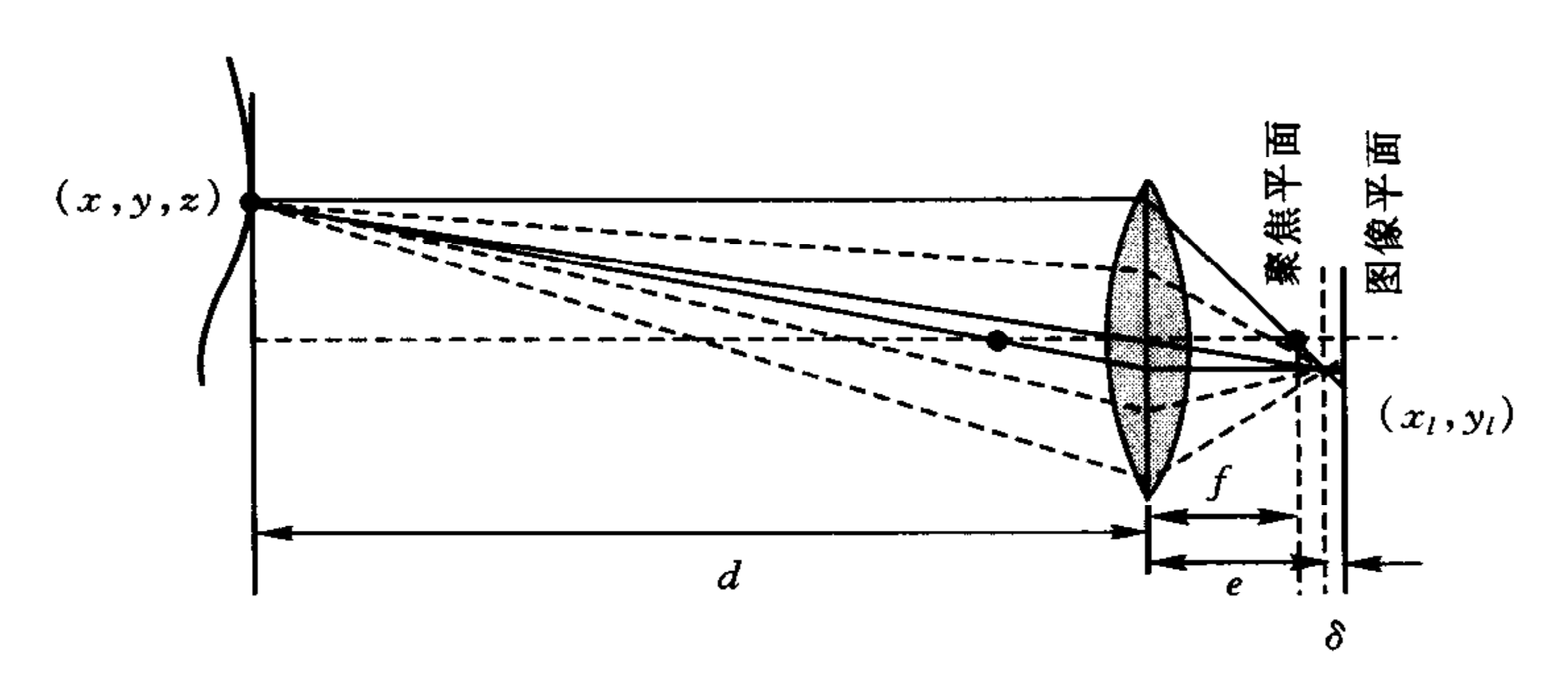


图 4.19 摄像机光学和它对图像影响的说明。为了得到锐利的图像，图像平面必须与聚焦平面重合，否则，如上图可见，点(x,y,z)的图像会在图像中变得模糊

支配图像形成的基本方程，建立了上图中从物体到镜头的距离 d 和从镜头到焦点的距离 e 的关系：

$$\frac{1}{f}=\frac{1}{d}+\frac{1}{e} \tag{4.19}$$

如果图像平面的位置与镜头的距离为 e，那么对于所示的特定物体的体素，全部的光将会聚焦在图像平面单独的点上，从而物体的体素会被**聚集**。但是，如果图像平面不在 e，如图 4.19 所示，则从物体体素来的光，会成为**模糊的圆**被投射到图像平面上。作为一次近似，假定光全部均匀地分布在该模糊圆上，而圆的半径 R 可以按下式表征：

$$R=\frac{L\delta}{2e} \tag{4.20}$$

L 是透镜或孔径的直径，δ 是距焦点的图像平面的位移。

给定这些方程，几个基本的光学效应就清楚了。例如，如果孔径或透镜缩成一个点，像一个针孔摄像机，则模糊圆圈的半径接近于零。这与事实相符：减少光圈孔径的开度，促使**景深**增加直至全部物体聚集。当然，这样做的缺点是，我们允许较少的光在图像平面上形成图像。所以，这实际上只在亮的环境里是可以的。

从这些光学方程可推论出来的第二个性质关系到模糊的灵敏度，它是透镜到物体的距离的函数。假定图像平面是在离直径为 $L=0.2$的透镜 1.2 的固定距离上，焦距 $f=0.5$。我们可以从方程(4.20)看到，模糊圆圈的尺寸 R 的变化正比于图像平面位移 δ。如果物体处在距离 $d=1$，则根据方程(4.19)我们可以计算出$e=1$，所以 $\delta=0.2$。增加物体距离到 $d=2$，结果 $\delta=0.533$。在各种情况下，利用方程(4.20)，我们可以分别计算 $R=0.02$ 和 $R=0.08$。这表明当物体接近透镜时，对散焦而言，灵敏度高。

相反，假定物体在 $d=10$ 的情况下，我们计算 $e=0.526$。但是，物体再移动一个单位到 $d=11$，则我们计算 $e=0.524$。最后的模糊圆半径为 $R=0.117$ 和 $R=0.129$。当障碍物距透镜的距离*是 1/10 时，R(的变化)远小于 4 倍。这个分析展示了景深的限制：当物体远离(给定固定的焦距)时，它们损失了灵敏度。令人感兴趣的是，这个

* 应是距离的变化——译者注。

限制实际上会最终加到所有视觉的测距技术上，包括立体深度和运动深度。

尽管如此，可以对计划中应用的深度范围，专门定制摄像机光学系统。例如，具有很大焦距 f 的变焦透镜，能在很大距离中赋予距离分辨率。当然，它以牺牲视场为代价。同样地，大的透镜直径与快速快门相配合会产生大的、更可检测的模糊圆。

给出由上面方程所概括的物理效应，人们可以想象一个采用多个图像的视觉测距传感器。在这种情况下，摄像机光学系统是变化的(例如，图像平面位移 δ)并捕捉到同一场景(图 4.20)。事实上，这种方法不是一个新的创造。人类视觉系统利用了丰富的暗示和技术，而且人类所展现的系统是深景的。人以大约 2 Hz 的速率连续地改变他们的焦距。为了使聚集最大化，这种方法主动地搜索透镜的光学系统，技术上称之为**聚焦深度(景深)**。相反，**散焦深度**意味着用一系列已被不同摄像机几何特征所拍摄的图像来恢复深度。

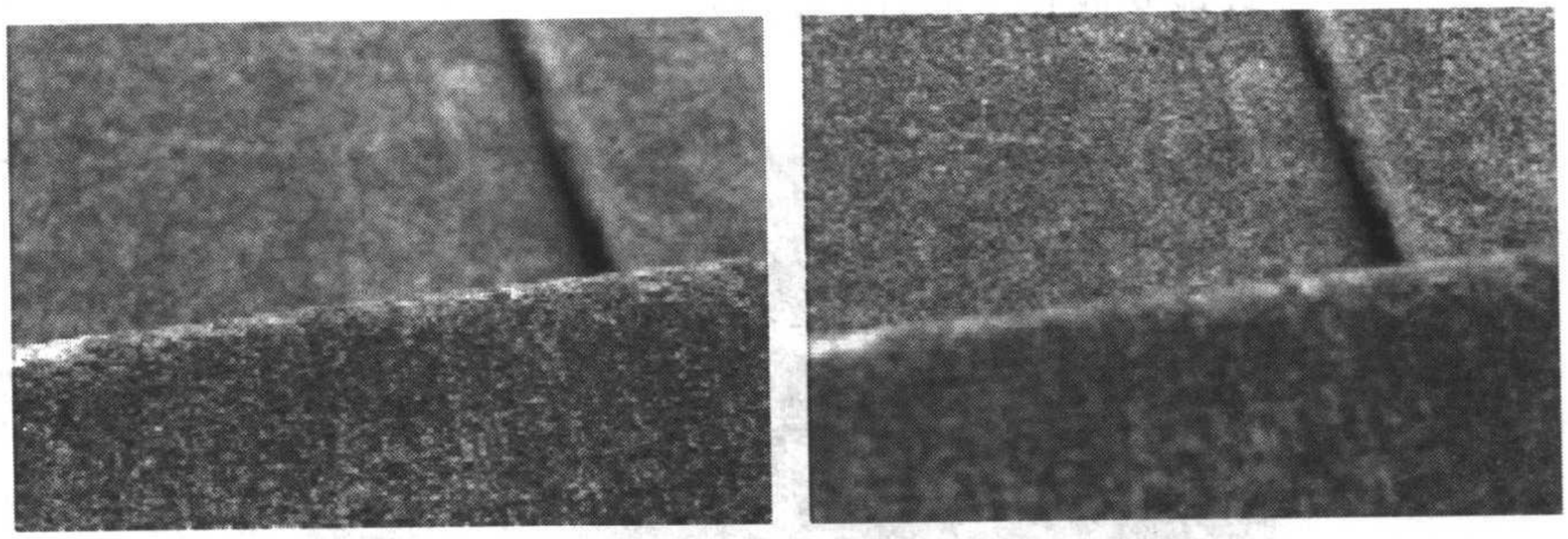


图 4.20 用摄像机在两个不同的位置拍摄的同一场景的 2 个图像。注意近表面和远表面之间纹理锐度的重大改变，场景是室外的水泥台阶

景深方法是最简单的视觉测距的技术之一。为了确定离物体的距离，传感器简单地将图像平面(通过聚焦)移动，直到将物体锐度最大化。当锐度被最大化时，图像平面的相应位置直接产生距离。某些自动对焦的摄像机和实际上所有视频摄像机都使用该技术。当然，需要有一个方法来测量图像的锐度或图像内的物体。最普通的技术是子图像的**强度(I)梯度**的近似测量：

$$锐度_1 = \sum_{x,y} | I(x,y) - I(x-1,y) | \tag{4.21}$$

$$锐度_2 = \sum_{x,y} (I(x,y) - I(x-2,y-2))^2 \tag{4.22}$$

差分技术(方程(4.21))的水平求和,其重要优点是,只要用一个整流器、一个低通滤波器和一个高通滤波器就行。计算可以用模拟电路实现。在商用摄像机和录像机中,这是一个普通的办法。这样的系统,沿着一个特殊的轴,对对比度敏感,但在实际条件下,这很少成为问题。

然而,景深方法是一个主动搜索,且会是缓慢的,因为它要花费时间。比如,利用伺服控制的聚焦环来改变摄像机聚焦参数。因此,该方法还未被用在移动机器人中。

景深技术的一种变形已经用于移动机器人中。它在各种不同的环境中演示避障和避开凹的障碍物(诸如台阶和边缘[117])。该机器人使用了 3 个单色摄像机,用不同的、固定的透镜聚焦位置,尽可能近地安放在一起(图 4.21)。

图 4.21 Cheshem 机器人使用 3 个单色摄像机作为它在人类环境中避障唯一的测距传感器,静态障碍物如灌木丛,凸障碍如边缘和台阶

所有 3 个帧同步的摄像机每秒几次同时捕捉同一场景的 3 幅图像。图像各分成 5 列和 3 行，或 15 个子区域。各区的近似锐度用方程(4.22)的变形计算，总共产生 45 个锐度值。注意，方程(4.22)是沿着对角但隔行地计算锐度的。这是由于一个巧妙但是重要的问题引起的。许多摄像机以隔行扫描的模式产生图像，这意味着首先捕捉奇数行，接着捕获偶数行。当在动态环境中使用这种摄像机时，例如，在移动机器人上，则相邻行表示 2 个不同的时间点(差别多至三十分之一秒)的动态场景，由于运动和非光学散焦，结果造成一个人为的模糊(图像混乱)。如果采用只比较偶数编号的行，则可以避免这个隔行扫描的副作用。

记住，3 个图像是用不同的焦点位置，各由一个摄像机拍摄的。根据焦点位置，我们分别称各图像为**近**、**中**、**远**图像。简单地比较 3 个相应区的锐度值，很快地构成了 3×5 场景的粗糙景深图像。因而，利用**近**、**中**、**远**的值，深度图像只把深度信息的二位分配给各区域。关键的一步是调整所有 3 个摄像机的焦点位置，使得障碍物前面的平地在深度图的行中产生中等读数。然后，不希望的，或是近的，或是远的读数会分别指明凹或凸的障碍物，使之能在地面的物体以及掉在地面上的物体的周围实现基本的避障。

虽然足以应付避障，但上面景深算法只提供不满意的、粗糙的距离信息。另一个选择是用**散焦深度**方法，它是最期望的基于视觉的技术。

散焦深度方法把同一场景的 2 个或多个图像作为输入，这些图像是用不同的已知摄像机几何特性拍摄的。给定图像和摄像机几何特征的设置，目标是恢复图像所表示的 3D 场景的深度信息。我们从推导实际场景性质(辐照度和深度)、摄像机几何设置和在图像平面组成的图像 g 之间关系开始。

场景的**聚焦图像** $f(x,y)$定义如下。考虑在透镜场所的一个针孔孔径($L=0$)，对在图像平面上的位置(x,y)的每一点 p，画一条经

孔径到实际场景相应的、可视的点 P 的直线。我们定义 $f(x,y)$ 为在 p 点由 P 来的光所形成的辐照度(或光强)。直觉上,$f(x,y)$ 表示理想聚焦的场景的强度图像。

点扩展函数 $h(x_g,y_g,x_f,y_f,R_{x,y})$ 定义为从场景 P 点(相应于 (x_f,y_f))到聚焦的图像 f 的辐照总量,在观察到的散焦图像 g 中,f 提供了点 (x_g,y_g)。注意,点扩展函数不仅取决于源 (x_f,y_f) 和目标 (x_g,y_g),也取决于 R。反过来,正如通过研究方程(4.19)和(4.20)可以知道的那样,模糊圆的半径依赖于从点 P 到镜头的距离。

假设模糊圆强度是均匀的,我们可以定义 h 如下:

$$h(x_g,y_g,x_f,y_f,R_{x,y})=\begin{bmatrix}\dfrac{1}{\pi R^2} & \text{如果}((x_g-x_f)^2+(y_g-y_f)^2)\leqslant R^2\\ 0 & \text{如果}((x_g-x_f)^2+(y_g-y_f)^2)>R^2\end{bmatrix} \tag{4.23}$$

直觉上,只有当点 P 的模糊圆包含点 (x_g,y_g) 时,点 P 才提供图像像素 (x_g,y_g)。现在我们可以写出计算图像 $f(x,y)$ 中各像素值的一般方程,它是点扩展函数和聚焦图像的函数:

$$g(x_g,y_g)=\sum_{x,y}h(x_g,y_g,x,y,R_{x,y})f(x,y) \tag{4.24}$$

这个方程,通过 R 建立了场景点深度对观察到的图像 g 的关系,求解 R 就向我们提供地图深度。但是该函数有另外的未知聚焦图像 f。所以,假如我们不知道全聚焦图像如何看,单独一个图像不足以解决深度恢复问题。

给定用不同摄像机几何特性拍摄的同一场景的两个图像,在理论上有可能求解 g 和 R,因为 f 保持恒定。存在许多可以准确快速完成这种解答的算法。经典的方法是所谓**逆滤波**,因为它力图直接求解 R,然后从这解中提取深度信息。逆滤波解答的一个特殊情况已经用一个实际的传感器予以证明。假如输入光被分裂并传送到两个摄像机,一个具有较大的孔径,另一个具有针孔的孔径[121],针孔的孔径产生全聚焦图像,直接提供 f 的值。用该方法,保留了一个

具有单个未知数的单个方程，所以解答是直接的。Pentland[121]已示范了这种传感器，它具有几米的量程，准确度高于97%。然而，要注意，针孔孔径需要大量的输入光，而且实际图像强度必须规格化，以使针孔和大直径图像具有等效的总辐射度。最近，散焦深度方法使用了统计技术，并把问题特征化为线性方程组[64]。这些基于矩阵的方法，近来在准确度方面超过所有以前的工作，取得了有重大意义的改进。

总之，散焦深度方法的基本优点是它的速度极快。上面的方程不像立体深度方法所面临的相关性问题那样，需要搜索算法以寻找解答。也许最重要的是，散焦深度方法也不需要获取不同视角的场景，所以不受阻塞和物体瞬间消失的影响。

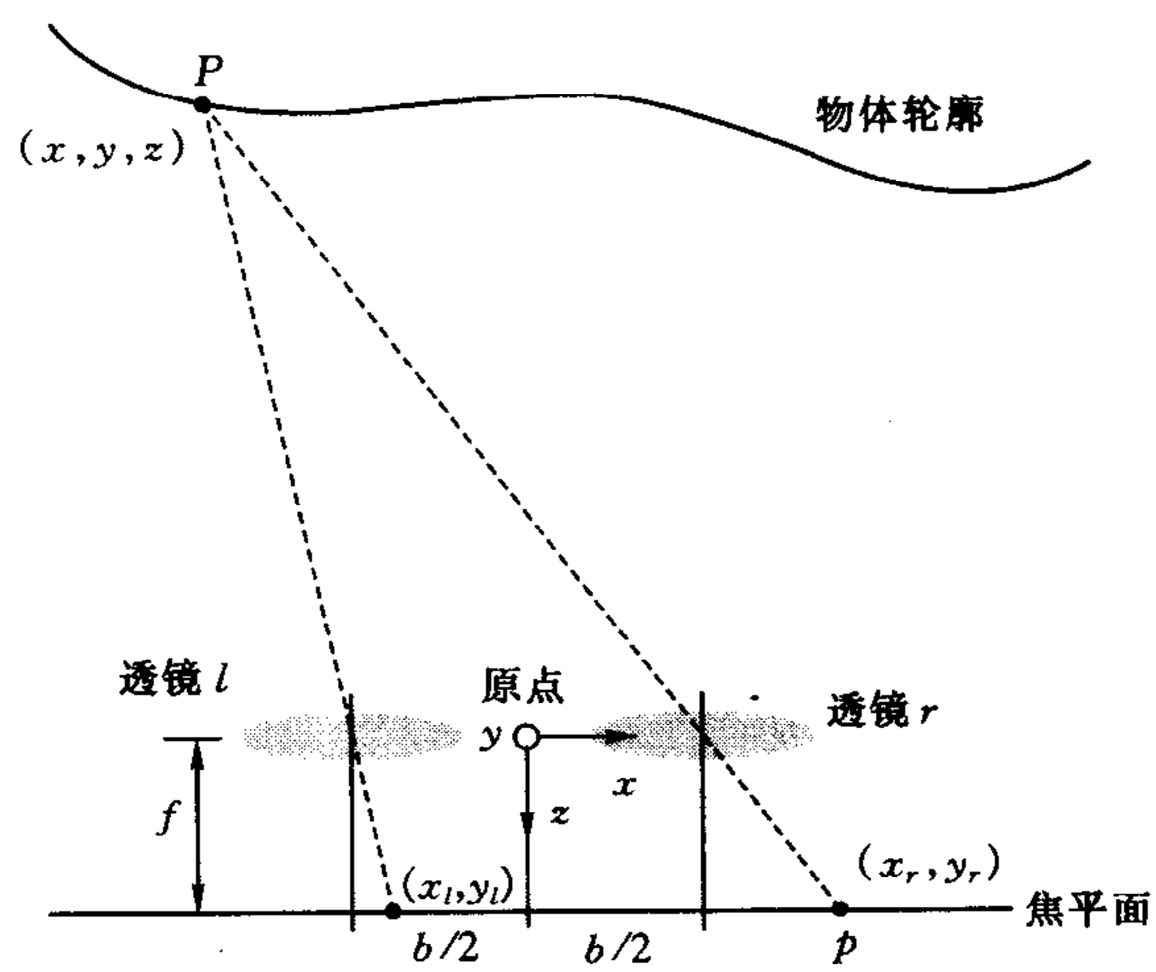


图 4.22 立体视觉理想化的摄像机几何特性

当用所有的视觉方法测距时，准确度随距离而降低。准确度确实可以是极大。这些方法已被应用于显微镜，以微米级展示测距。

立体视觉 立体视觉是我们从不同视角刻画的两个图像，恢复深度信息的几个技术之一。多年来，我们已经充分理解了立体深度的理论，而创建一个实际的立体传感器，其工程上的挑战却一直是令

人生畏的[16,29,30]。在这前沿，近来已经看到了第一个成功的结果。所以，在提出一个立体测距的基本数学方程之后，我们描述最高技术水平的算法和最近商业上可提供的一个立体传感器。

首先，我们考虑一个简单的情况，这里，如图 4.22 所示，在间距 b 处（称作**基线**）安放了两个摄像机，它们的光学轴平行。

在该图中，物体上的一个点被描述为处在相对于中心原点的坐标(x,y,z)，而原点位于两个摄像机的透镜之间。在各摄像机图像上，该点的光信号位置以摄像机特定的局部坐标表示。因此，形式为(x_l,y_l)的点，其所参考的坐标系的原点位于透镜 l 的中心。

从图 4.22，我们可以看到

$$\frac{x_l}{f}=\frac{x-b/2}{z},\ \frac{x_r}{f}=\frac{x-b/2}{z} \tag{4.25}$$

和（超出页平面）

$$\frac{y_l}{f}=\frac{y_r}{f}=\frac{y}{z} \tag{4.26}$$

式中 f 是两个透镜到图像平面的距离。注意，从方程(4.25)得到

$$\frac{x_l-x_r}{f}=\frac{b}{z} \tag{4.27}$$

这里，图像坐标中的差 x_l-x_r，被叫做**不一致性**。在立体视觉中，这是一个重要的项，因为只有通过测量不一致性我们才能恢复深度信息。利用不一致性并解上面 3 个方程，产生被摄图像的场景点的三维方程：

$$x=b\frac{(x_l-x_r)/2}{x_l-x_r};\ y=b\frac{(y_l-y_r)/2}{x_l-x_r};\ z=b\frac{f}{x_l-x_r} \tag{4.28}$$

从这些方程，观察如下：

- 距离反比于不一致性。所以，正如用景深技术一样，近距离的物体比远距离物体可测得更准确。一般，这对移动机器人是可接受的，因为对导航和避障而言，比较接近的物体更重要。
- 不一致性正比于 b。对一个给定的不一致误差，深度估计的准确度随基线 b 的增加而提高。

- 当 b 增加时，因为摄像机之间物理间距增加，某些物体也许在一个摄像机出现，而在另一个没有出现。这种物体按定义将不会有不一致性，所以测距将不会成功。
- 对两个摄像机都可视的场景中的点，产生一对图像点（经各透镜），称为**共轭对**。给定共轭对的一个成员，我们就知道共轭对的另一成员处在沿着一条称之为**表偏振线**的直线的某一地方。在图 4.22 所示的情况下，因为摄像机相互理想地排列成一行，表偏振线就是水平线（即沿 x 方向）。

但是，理想的排列成行的摄像机这种假定在实际中常常受到破坏。为了优化可被恢复的距离范围，将摄像机向内转向另一个常常是有用的。图 4.22 表示了需要解决这个更一般问题的**方向**向量。我们将分别根据各摄像机的参考框架，表述场景点 P。摄像机的参考框架不必排列成行，而且确实可以相对于另一个的任何方向。

例如，点 P 的位置根据左边摄像机框架描述成 $r'_l=(x'_l, y'_l, z'_l)$。注意，这些是点 P 的坐标，不是左边摄像机图像中它的对应物体的位置。P 同样可以根据右边摄像机框架描述成 $r'_r=(x'_r, y'_r, z'_r)$。如果我们用一个转动矩阵 R 和平移矩阵 r_0，将摄像机 l 和 r 的相对位置关联起来，则我们根据 r'_l，确定 r'_r：

$$r'_r = R \cdot r'_l + r_0 \tag{4.29}$$

式中 R 是 3×3 转动矩阵，r_0 是两个摄像机之间偏置平移矩阵。

将方程(4.29)扩展，得到

$$\begin{bmatrix} x'_r \\ y'_r \\ z'_r \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x'_l \\ y'_l \\ z'_l \end{bmatrix} + \begin{bmatrix} r_{01} \\ r_{02} \\ r_{03} \end{bmatrix} \tag{4.30}$$

上面方程有两个用途：

(1) 如果我们知道 R，r'_l，和 r_0，我们可求得 r'_r。当然，如果我们知道 r'_l，我们会有相对于左边摄像机关于 P 位置的完整信息，所以解决了深度恢复问题。注意，如图 4.22 那样，对理想地排列成行的

摄像机，$\boldsymbol{R}=\boldsymbol{I}$(单位矩阵)。

(2) 给定一组共轭对$\{(x_l', y_l', z_l'), (x_r', y_r', z_r')\}$，我们可以校准系统，并找到$r_{11}, r_{12}\cdots$。

为了实现上面步骤2的校准步骤，我们必须求解12个未知的值，需要12个方程。这意味着对给定的场景，校准需要4个共轭对。

上面的例子假定，对用两个摄像机的立体深度恢复，都要求正规的平移和转动以使校准充分有效。事实上，单个摄像机的校准本身是一个研究的活跃领域。特别是当目标包括任何3D恢复图像时。当研究人员想用3D高精度的单个摄像机时，相对于透镜光轴，必须对有关成像芯片的精确安放的内部误差以及透镜系统本身的失常予以校准。这种单摄像机的校准，涉及相对于光轴(平移和角度)成像芯片精确偏移值的求解，以及沿着成像芯片的表面和外部受观察的表面寻求距离之间的关系。而且，即使没有拍摄中的光学失常，由于透镜本来就是一个光射仪器，所以投影到平坦成像表面的图像是放射地畸变(即平行线在受观察的世界中收敛到成像芯片)的。

这种单摄像机校准的一个公用的实际技术，是基于一个易于分析的平面模式获取多个视图，诸如在白背景上的黑方块栅格。可容易地提取这些方块的拐角，并利用交互式的细化算法，可以提取摄像机内在的校准参数。因为现代成像系统有远远超过像素尺寸的空间准确度的能力，所以这种精细校准的性能指标具有重要的意义。校准的进一步讨论和下载，以及使用标准校准程序的方法可参阅参考文献[158]。

假定校准步骤是完整的，我们现在可以对距离恢复问题建立方程。开始我们没有可用的P位置，所以，(x_l', y_l', z_l')和(x_r', y_r', z_r')不知道，代之以依靠两个摄像机，我们有各摄像机图像平面上的像素，(x_l, y_l, z_l)和(x_r, y_r, z_r)。给定摄像机的焦距f，我们可以把P位置对左边摄像机图像建立如下关系：

$$\frac{x_l}{f}=\frac{x_l'}{z_l'}, \quad \frac{y_l}{f}=\frac{y_l'}{z_l'} \tag{4.31}$$

让我们首先集中在值 z'_l 和 z'_r 的恢复，从方程(4.30)和(4.31)，我们可以从下面任意两个方程计算这些值：

$$\left(r_{11}\frac{x_l}{f}+r_{12}\frac{y_l}{f}+r_{13}\right)z'_l+r_{01}=\frac{x_r}{f}z'_r \tag{4.32}$$

$$\left(r_{21}\frac{x_l}{f}+r_{22}\frac{y_l}{f}+r_{23}\right)z'_l+r_{02}=\frac{x_r}{f}z'_r \tag{4.33}$$

$$\left(r_{31}\frac{x_l}{f}+r_{32}\frac{y_l}{f}+r_{33}\right)z'_l+r_{03}=z'_r \tag{4.34}$$

相同的过程可以识别 x' 和 y'，获得关于点 P 位置的完整信息。然而，利用上面的方程，要求我们辨识左边和右边摄像机图像中的共轭对：源于场景同一物体点 P 的图像点。这个基础性的挑战，即辨识共轭对和由此恢复不一致性，是一个**对应性问题**。直觉上，问题是：给定从不同角度同一场景的两个图像，我们如何能辨识在两个图像中相同的物体点？对每一个这种被辨识的物体点，我们有可能恢复场景中它的 3D 位置。

对应性问题或者两个不同输入匹配同一物体的问题，已成为计算机视觉领域和人工智能领域最具挑战性问题之一。为了创建更稳定、更丰富的信息数据，在几乎所有提议的解决方案中都涉及转换各个图像。有了更可靠的数据在手，立体算法则是**搜索**代表尽可能多的图像像素的最佳共轭对。

搜索过程很好理解，但是最终的深度图的质量很大程度上依赖于缩减噪声和改善稳定性的图像处理方法。这已成为立体视觉算法首要的技术驱动力，而且有一个特别的方法已在商业现有的系统中被广泛使用。

高斯的拉普拉斯零相交(ZLoG) ZLoG 是辨识左和右摄像机图像特征的一个策略，它稳定且匹配良好，能得到高质量的立体深度恢复。该方法在立体视觉领域已获得巨大成功，以优良的成果在软件和硬件两方面已经实现了商业化。它已产生了几个商业立体视觉系统，尽管它极其简单。这里我们概述其方法并阐明它的某些优点。

ZLoG 的核心是图像的拉氏变换。直觉上，这不过是二次导数。形式上，具有强度 $I(x,y)$ 的一个图像拉普拉斯 $L(x,y)$ 定义为：

$$L(x,y)=\frac{\partial^2 I}{\partial x^2}+\frac{\partial^2 I}{\partial y^2} \tag{4.35}$$

所以，拉普拉斯表示图像的二次导数，并沿两个轴计算。这种变换叫做**卷积**，必须对像素值的离散空间进行计算。为了应用，要求解方程(4.35)的近似：

$$L=P\otimes I \tag{4.36}$$

我们描述离散算子 P，称为**核**。沿着两个轴，它把二次导数运算近似为一个 3×3 的表：

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.37}$$

应用核 P，卷积图像是直截了当的。核定义了图像中各像素对目标中相应的像素和它的邻居的贡献。例如，如果一个图像 I 中一个像素(5,5)具有值 $I(5,5)=10$，则应用方程(4.37)所描述的核，像素 $I(5,5)$ 对目标图像的贡献如下：

$$L(5,5)+=-40;$$
$$L(4,5)+=10;$$
$$L(6,5)+=10;$$
$$L(5,4)+=10;$$
$$L(5,6)+=10。$$

现在考虑一个阶跃函数的图例，阶跃函数表示强度为黑的像素行，突然跳到很亮的强度。如图 4.23 所示，二次导数会有一个陡的正峰值，接着是一个陡的负峰值。用拉氏是因为图像对变化的极端敏感性。但是事实上二次导数是过敏感的，我们希望由于场景强度的实际变化，拉普拉斯引发大的峰值，但我们希望抑制引发假峰值的信号噪声。

为了去除由于传感器误差造成的噪声，ZLoG 算法首先施加高

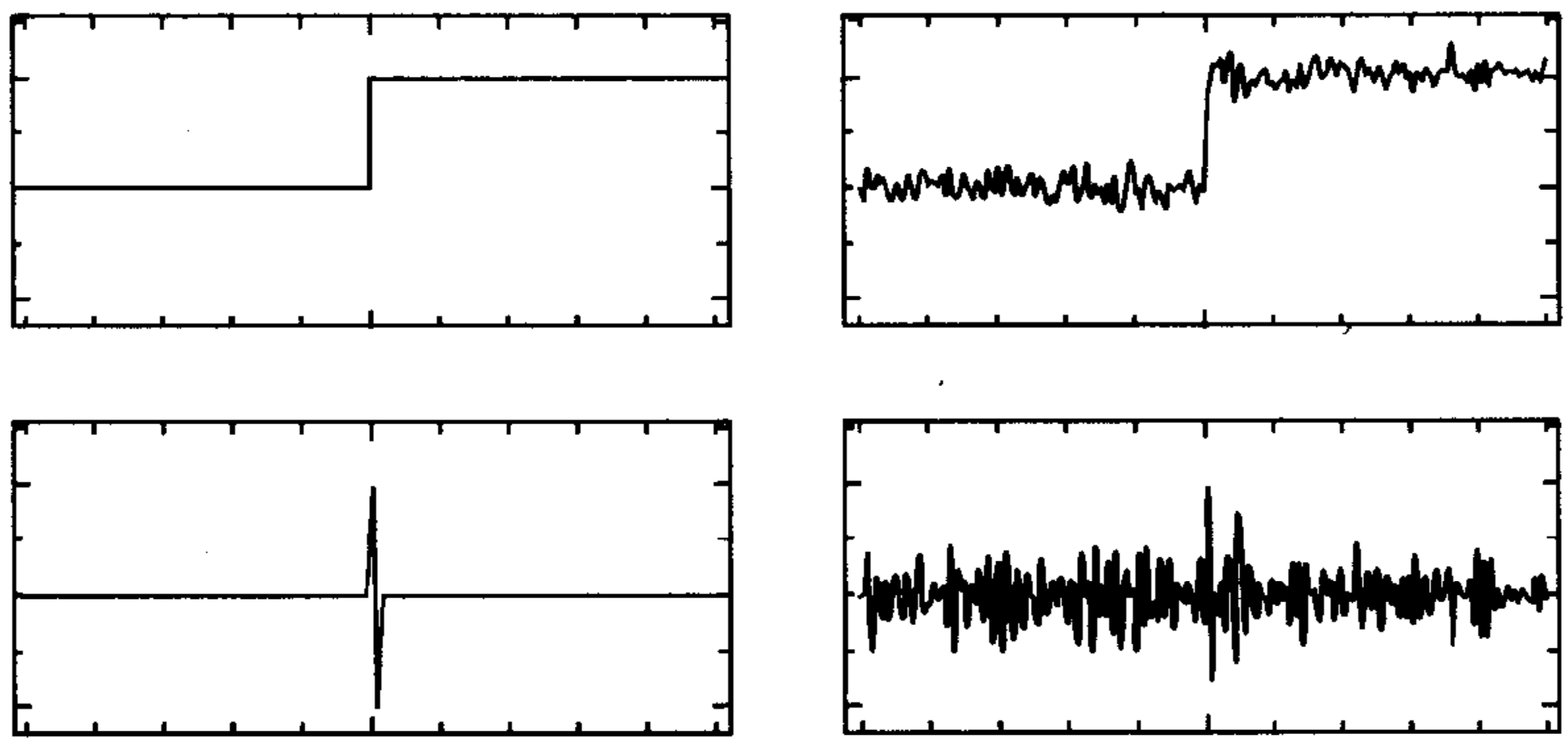


图 4.23 二次导数的阶跃函数实例和噪声的影响

斯平滑，然后执行拉氏卷积。通过与近似高斯平滑的 3×3 表的卷积，这种平滑可以是有效的：

$$\begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix} \tag{4.38}$$

高斯平滑并不真正消除误差，它仅仅将图像变动分布到更大的区域。这似乎应是熟悉的。事实上，高斯平滑几乎与散焦光造成的模糊是相同的。但是，正像最小模糊消除细密纹理的细节一样，这在消除高频噪声是非常有效的。注意，像散焦一样，这个核并不改变总的照明，但仅仅将它重新分布(依靠除数 16)。

高斯图像滤波的拉氏变换，其结果是一个辨识原始图像变化的边界和具有正、负陡的尖峰脉冲的目标阵列。例如，图像中陡的边缘会在边缘任一边产生既是正的又是负的尖峰脉冲。

为了解决对应性问题，我们愿意辨识 LoG 的特定特征，它能够修正左和右摄像机的被滤波图像之间的匹配。一个非常有效的特征是把 LoG 的零交叉辨认为这种特征。在图像的边缘确实存在许多

零交叉，但它们的发生多少要比边缘更宽。零交叉的一个有意义的特性是它们非常锐利地被定义，即在被滤波的图像中只覆盖一个“像素”。利用内插法，用子像素的准确度建立零交叉的位置，准确度甚至可进一步提高。考虑到所有情况，ZLoG 零交叉的准确度已经使它们在目前技术发展水平中成为令人更喜欢的特征。

图 4.24 表示了一个实例，说明从立体图像提取深度信息所需的不同的步骤。

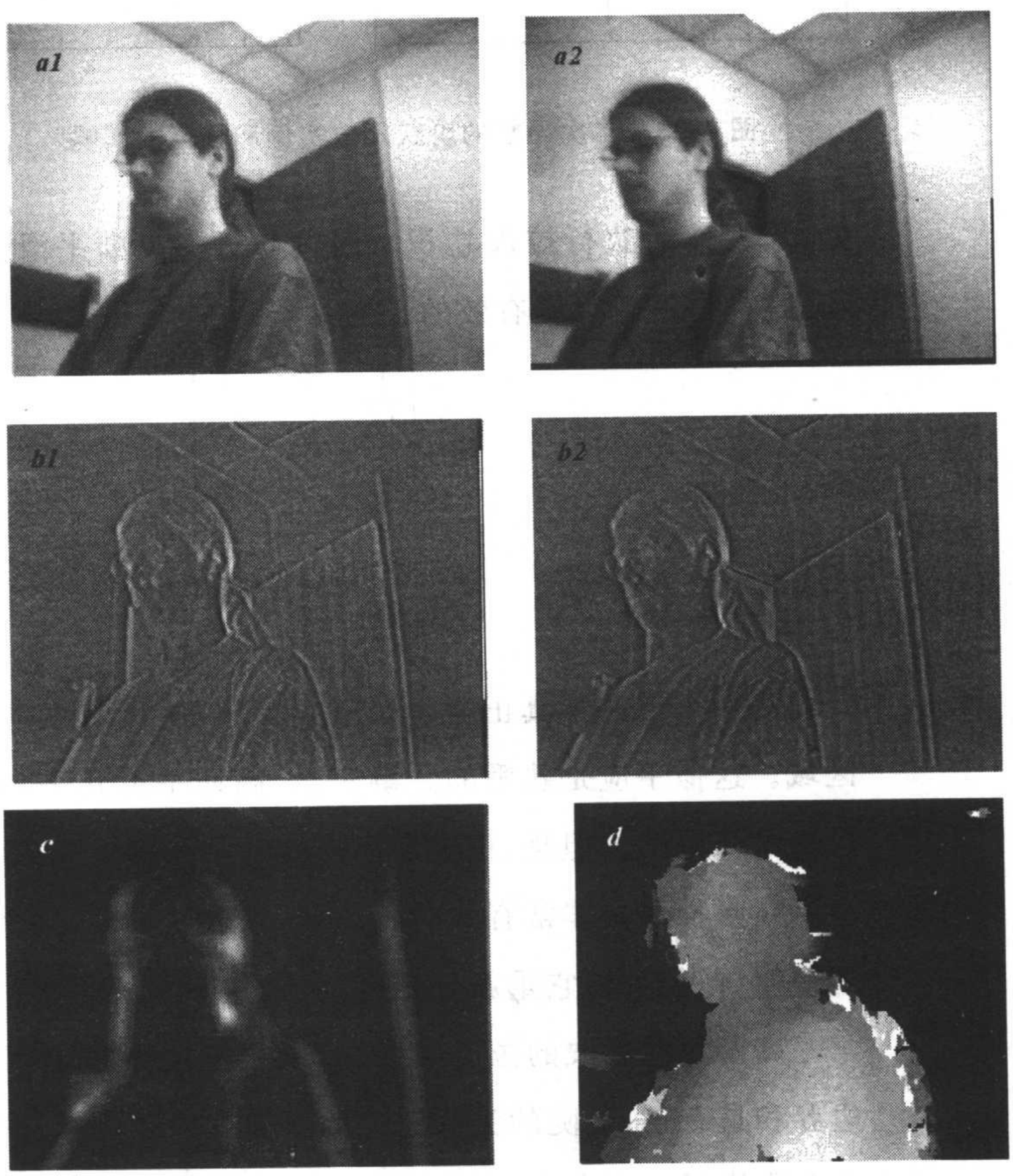


图 4.24 从立体图像中提取深度信息。(a1)和(a2)分别是左和右图像；(b1)和(b2)分别是垂直边缘滤波的左和右图像：滤波器＝[1 2 4 −2 −10 −2 4 2 1]；(c)置信图像：亮＝高置信度(纹理良好)；黑＝低置信度(无纹理)；(d)深度图像(不一致性)：亮＝近；黑＝远

在过去的10年里，已经为研究人员提供了几个商用立体视觉深度恢复的传感器。今天，在移动机器人学中，一个流行的单元表示在图4.25中，Videre Design公司的数字立体头(或SVM)。

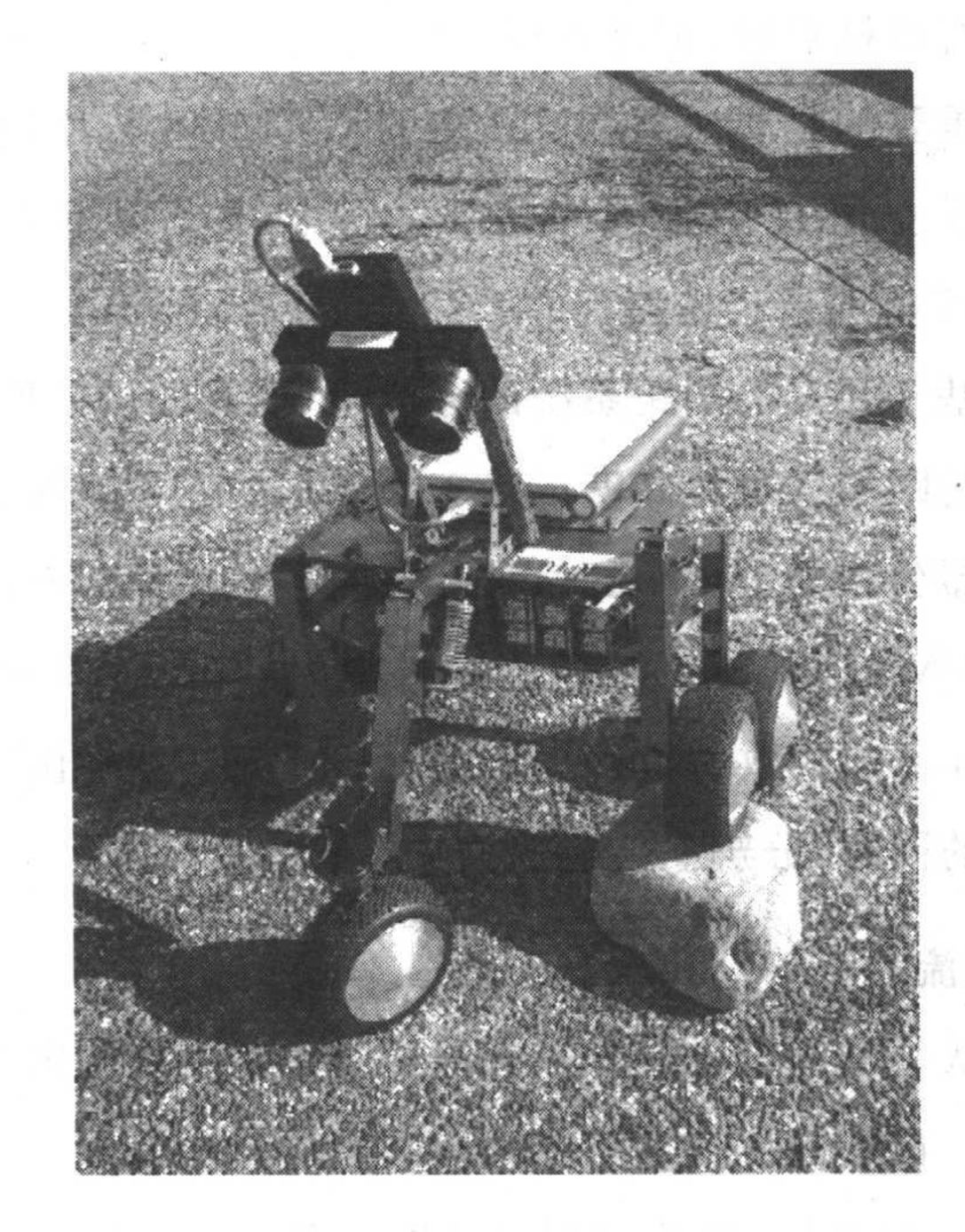


图4.25 装在EPFL的Shrimp机器人的SVM模块

SVM使用LoG算子，随之，将最终的阵列棋盘格化成子区域，在子区域内，计算绝对值之和。在这些子区域层次上解决了对应性问题，这被称为**区域相关性**过程。对应性问题解决之后，结果被内插成四分之一像素精度。SVM的一个重要特征是，它产生不仅是深度图，而且是各像素匹配质量的度量。这是有价值的，因为这种附加的信息随时可用于消除虚假的、不正确的、匹配质量差的立体匹配。

SVM的特性指标提供了当前立体测距技术发展水平的典范。SVM由传感器硬件组成，包括两个CMOS摄像机和DSP(数字信号处理)硬件。另外，SVM包括使用标准计算机(例如奔腾处理器)的立体视觉软件。在一个320×240像素图像对上，SVM以每秒12帧

速率(基于奔腾 II 233 MHz),将 1/32 离散的不一致性(即深度)电平分配到每一个像素,这比激光测距仪和超声更令人喜欢。特别当人们鉴赏到立体测距信息不仅仅为一个目标点的计算,而是为图像中所有目标点时,更是这样。

重要的是要注意,SVM 用的是 CMOS 芯片,不是 CCD 芯片,它展示了立体视觉算法有足够的分辨率。用不太贵的和功效好的 CMOS 技术是容易实现的。

基于视觉的测距系统的分辨率,如我们以前所述那样,依赖于距物体的距离。虽然高度依赖于摄像机的光学系统,但考察一下 SVM 传感器公布的分辨率的值是有益的。使用一对标准的 6 mm 焦距透镜,SVM 宣称,3 m 距离的分辨率为10 mm;10 m 距离的分辨率为 60 mm。这些数值虽都基于理想的环境。却举例说明了基于视觉的测距的同时,分辨率会迅速损失。

4.1.8.3 运动和光流

从一个固定(或移动)的摄像机记录时变的图像可以恢复大量的信息。首先,我们区分运动场和光流的差别:

- 运动场:将速度向量分配给图像中的每一点。如果环境中一个点以速度 v_0 移动,则在图像平面引起速度 v_i。可以在数学上确定 v_0 和 v_i 之间关系。
- 光流:这也是真实的,图像中亮度模式随着促使它们移动的物体(光源)而运动。光流是这些亮度模式的视在运动。

在我们的分析中,我们假定光流模式将对应于运动场。虽然,在实际中并不经常如此。图 4.26(a)说明了该情况。图中球体表示亮度或阴影在球体图像中的空间变化,因为球体的表面是弯曲的。然而,如果表面运动,这个阴影模式将不动。因此,即使运动场不是零,光流到处是零。在图 4.26(b),相反的情形发生了。这里我们有一个具有动光源的固定球体。图像中的阴影随光源运动而改变。在这情况下,光流是非零,但运动场是零。如果我们可获取的信息只是光

流，且我们依靠它，那么在这两种情况下，我们都会得到不正确的结果。

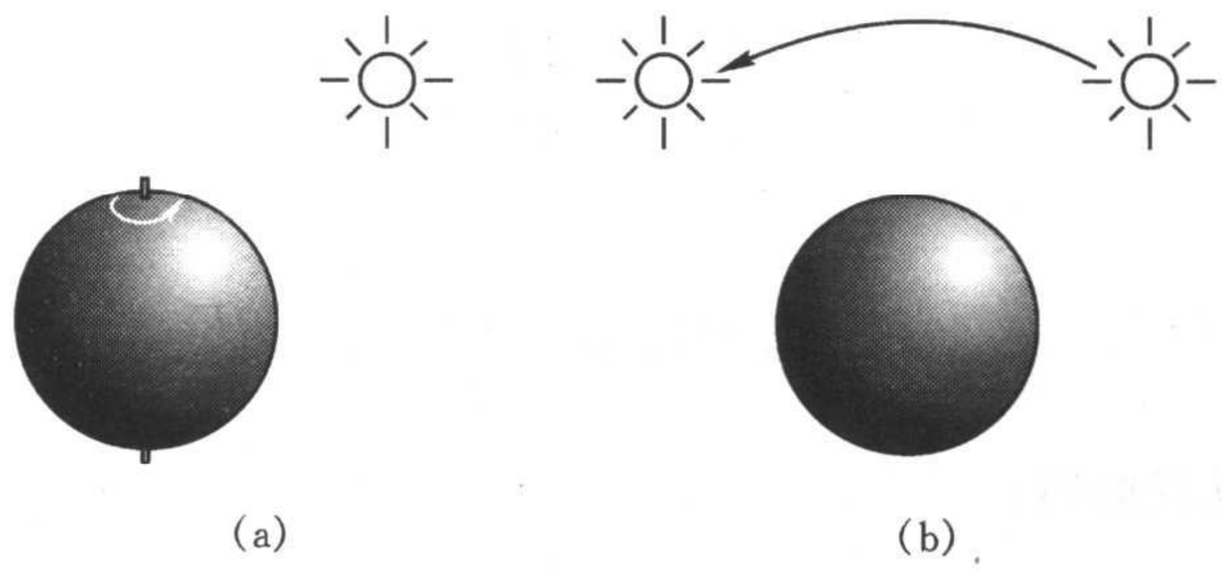


图 4.26 图中，球体或光源的运动证明光流不总是同运动场一样

光流 想测量光流并由此得到场景的运动场有许多技术。大多数算法利用局部信息，力图在两个相继的图像中寻求局部光斑的运动。在某种情况下，有关平滑度和一致性的全局信息可以帮助进一步消除这种**匹配**过程的歧义。下面我们提出光流约束方程法的细节。关于这方面更详细资料和其他方法，可参考文献[41,77,146]。

首先，假定相继快照之间的时间间隔是如此的快，以至于我们可以认为同一物体部分的被测强度实际上是恒定的。数学上，令 $E(x,y,t)$ 为 t 时刻在图像点 (x,y) 上图像的辐射照度。如果 $u(x,y)$ 和 $v(x,y)$ 是那点光流向量的 x 和 y 分量，则我们需要在时刻 $t+\delta t$，为辐射照度是相同的点，即，点 $(x+\delta t, y+\delta t)$，搜索一个新图像，式中 $\delta x = u\delta t$ 和 $\delta y = v\delta t$。即，对小的时间间隔 δt，

$$E(x+u\delta t, y+v\delta t, t+\delta t) = E(x,y,t) \tag{4.39}$$

通过时间 t，这会获得恒定强度**斑点**的运动。如果我们进一步假定图像亮度变化平滑，则我们可把方程(4.39)的左边，展开成泰勒级数，得到：

$$E(x,y,t) + \delta x \frac{\partial E}{\partial x} + \delta y \frac{\partial E}{\partial y} + \delta t \frac{\partial E}{\partial t} + e = E(x,y,t) \tag{4.40}$$

式中 e 包含 δx 中二阶和更高阶的项，等等。在极限情况下，当 δt 趋向零，我们得到

$$\frac{\partial E}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t}+\frac{\partial E}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t}+\frac{\partial E}{\partial t}=0 \tag{4.41}$$

上式可缩写为

$$u=\frac{\mathrm{d}x}{\mathrm{d}t};\ v=\frac{\mathrm{d}y}{\mathrm{d}t} \tag{4.42}$$

和

$$E_x=\frac{\partial E}{\partial x};\ E_y=\frac{\partial E}{\partial y};\ E_t=\frac{\partial E}{\partial t}=0 \tag{4.43}$$

所以我们得：

$$E_x u+E_y v+E_t=0 \tag{4.44}$$

导数 E_t 表示强度如何快速地随时间而变，而导数 E_x 和 E_y 表示强度变化的空间率（强度变化如何快速地跨越图像）。合在一起，方程(4.44)就被称为**光流约束方程**，且给定相继的图像，对各个像素，可以估计出 3 个导数。

对各像素，我们需要计算 u 和 v 两个量。但是光流约束方程只为每一个像素提供一个方程，所以这是不充分的。当人们考虑许多等强度像素本质上可以是含糊时，其不确定性直觉上是显然的——也许不清楚的是在先验图像中，对一个等强度的原始像素，哪一个像素是最终的位置？

解决这个不确定性需要附加的约束。我们假定，一般而言，相邻像素的运动是相似的，因此全部像素的总光流是平滑的。这个约束是有意义的，因为我们知道它受到某种程度的破坏。为了使光流计算易于处理，我们仍然加强约束。特别当场景中不同物体相对于视觉系统按不同方向运动时，该约束正好被破坏。当然，这种情况会趋向于包括边缘，且可能引入一个有用的可视提示。

因为我们知道该平滑性约束有些不正确，所以我们可以在数学上通过计算以下的方程，定义我们破坏该约束的程度

$$e_s=\iint(u^2+v^2)\mathrm{d}x\mathrm{d}y \tag{4.45}$$

它是光流梯度幅值平方的积分。我们也可确定光流约束方程中的误

差(实际上不会正好是零)。

$$e_c = \iint (E_x u + E_y v + E_t)^2 \mathrm{d}x\mathrm{d}y \tag{4.46}$$

这两个方程应尽可能地小,所以我们要使 $e_s + \lambda e_c$ 最小,式中 λ 是一个参数,它在相对于偏离平滑的图像运动方程中,对误差加权。如果亮度测量准确,应使用大的参数;如果它们是噪声,则用小参数。在实践中,手动地和交互地调整参数 λ,以达到最佳的特性指标。

因此,最后的问题实际上是变分计算,得到欧拉方程:

$$\nabla^2 u = \lambda(E_x u + E_y v + E_t)E_x \tag{4.47}$$

$$\nabla^2 v = \lambda(E_x u + E_y v + E_t)E_y \tag{4.48}$$

式中

$$\nabla^2 = \frac{\partial^2}{\delta x^2} + \frac{\partial^2}{\delta y^2} \tag{4.49}$$

这是拉普拉斯算子。

方程(4.47)和(4.48)组成一对可以被重复求解的椭圆二阶偏微分方程。

哪里发生侧影(一个物体挡住另一物体),哪里就会在光流中产生断续。这当然破坏了平滑性的约束。一个可能性是努力找到预示这种阻挡的边缘,从光流计算中排除这种边缘附近的像素,使得平滑性成为更现实的假设。另一个可能性是随机地利用这些独特的边缘。事实上,拐角可特别容易地跨越后续的图像进行**模式-匹配**,因此凭着它们本身的能力,可以用作光流计算的基准标记。

在未来跨多种算法的联合提示的视觉算法中,光流计算有希望成为的一个重要成分。但是,还没有证明,专门使用光流的移动机器人,其避障和导航控制系统是广泛有效的。

4.1.8.4 颜色跟踪传感器

虽然立体深度无疑将证明,它是移动机器人学基于视觉的一个普遍应用的方法,但它模仿了现有传感器的机能,包括超声、激光和光学测距仪。基于视觉感知的一个重要方面是视觉芯片可以提供其

他移动机器人不能提供的感知模式和提示。一个这种新感知的模式是检测和跟踪环境中的颜色。

颜色表示了正交于距离的环境特征，而且它既表示了自然提示，又表示了人工提示，它可向移动机器人提供新的信息。例如，每年机器人足球赛为了环境标记和足球定位广泛使用了颜色(图 4.27)。

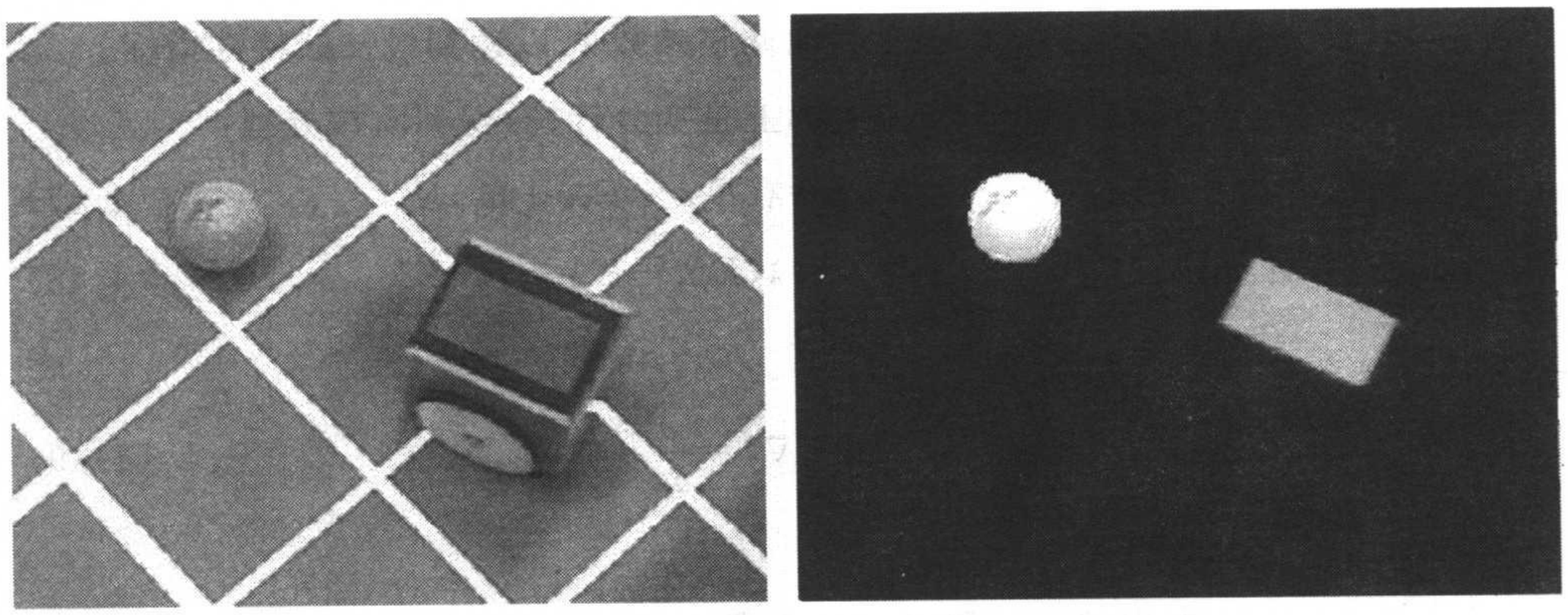


图 4.27 EPFL 的 STeam Engine 足球机器人顶部的颜色标记能使颜色跟踪传感器将足球场中的机器人和足球定位

颜色感知有两个重要的优点。第一，颜色检测是单个图像的一个简单函数。所以，在这种算法里不需求解对应性问题。第二，因为颜色感知提供了新的、独立的环境提示，如果它与现有的提示结合，诸如立体视觉或激光测距仪数据，我们可以期待有很大的信息增益。

有效的颜色跟踪传感器现在商业上已可提供。下面，我们简要地描述两个商用的、基于硬件的颜色跟踪传感器和普遍可用的基于软件的解决方案。

Cognachrome 颜色跟踪系统 牛顿研究实验室的 Cognachrome 视觉系统是一个基于硬件的颜色跟踪传感器，能在专用处理器上[162]以极快速度跟踪颜色。系统根据用户定义的颜色，以 60 Hz 速率检测色斑。Cognachrome 系统最大可对每帧 25 个物体作检测和报告，对各物体独立地提供质心、界定的框体、面积、纵横比和主轴方向等信息。

该传感器利用一个称作**不断设限**的技术来辨识各颜色。在RGB(红、绿、蓝)空间中,用户对各个红、绿、蓝定义最小和最大值。由这6个约束所定义的3D框体形成一个颜色界定框,所有在该界定框内具有RGB值的任何像素被定义为一个目标。目标像素被并入到较大的向用户报告的物体。

Cognachrome传感器在一个规模为200×250像素的画面中,对各物体的质心,其位置分辨率达到一个像素。该传感器的重要优点,如激光测距仪和超声一样,是由于感知的模式在移动机器人的主处理器上无负载。全部的处理是在传感器专用的硬件(即摩托罗拉68332和配对的帧捕捉器)上进行。Cognachrome系统价值几千美元,但正在被牛顿实验公司更高性能的硬件视觉处理器所替代。

CMUcam机器人视觉传感器 近来,芯片生产的进展,既在CMOS成像传感器方面又在50^{+} MHz范围的高速、易于获得的微处理器方面取得了进展,已经使它能制造出低开销的、与Cognachrome功能相似的智能视觉传感器,但价格只是它的小数。CMUcam传感

图4.28 由3个芯片组成的CMUcam传感器:CMOS成像芯片;SX28微处理器;和Maxim RS232电平转换器[126]

器是一种新的系统,如图4.28所示,它与带有专用CMOS成像芯片的低价微处理器配对,产生智能的、整装的视觉传感器,价格为100美元。

设计该传感器，用来把从摄像机图像提取的高级信息提供给一个外处理器。举例来说，它也许可控制一个移动机器人。外处理器配置了传感器的流动数据模式，比如，对有限的 **RGB** 或 **YUV** 的数据集合，指定其跟踪模式，然后视觉传感器实时处理数据并输出高级信息给外界用户。该传感器以小于 150 mA 的驱动电流，在分辨率为 80×143[126] 的情况下，以每秒近似 20 帧的速度提供图像颜色统计和颜色跟踪服务。

图 4.29 展示了一旦传感器按人手进行训练后，由 CMUcam 提

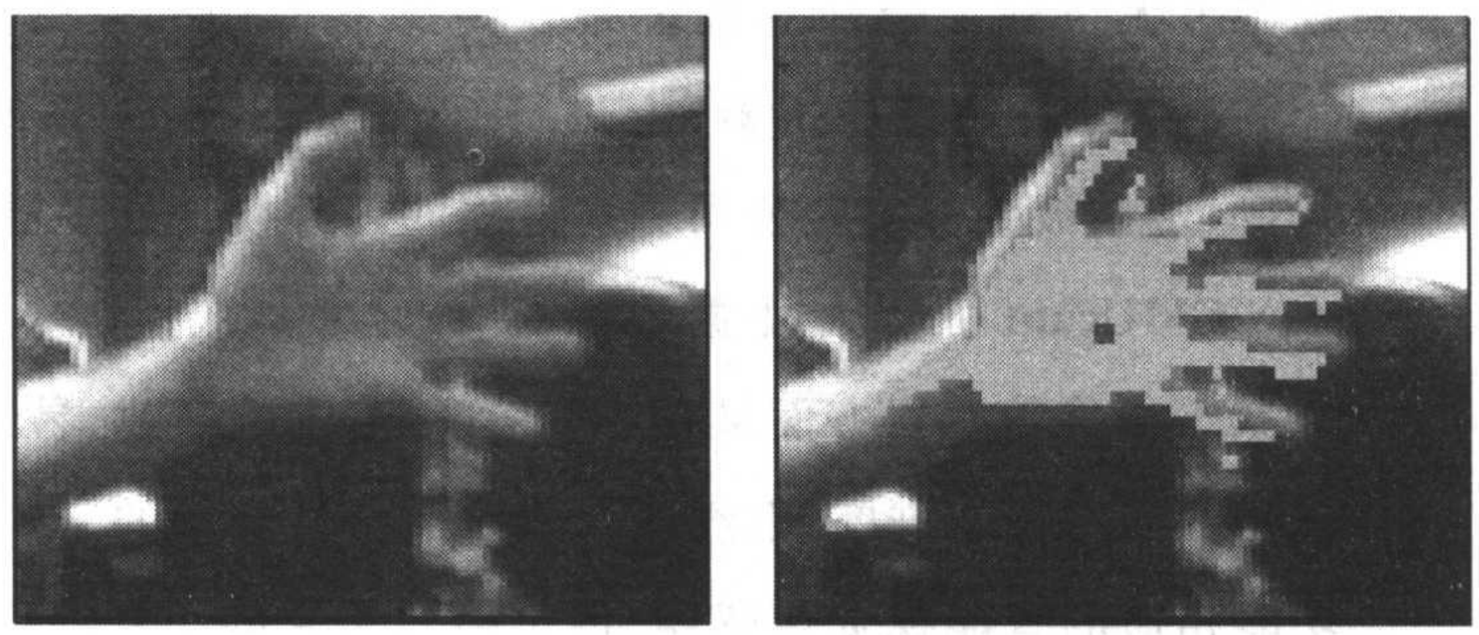


图 4.29 用于人手时基于颜色的物体提取

供的基于颜色的物体的跟踪服务。它提取了物体的近似形状，以及它的定界框体和质心。

CMVision 颜色跟踪软件库 近来，因为处理器迅速加速，在移动机器人内部的主处理器上有执行基本视觉处理的趋势。英特尔公司的计算机视觉数据库正是这种处理的优化数据库[160]。在这方面，CMVision 视觉跟踪软件代表了在动态环境中颜色跟踪的软件解决方案的最新技术水平[47]。CMV 在标准 200 MHz 奔腾计算机上，可以以 30 Hz 跟踪高达 32 种的颜色。

该传感器所用的基本算法，像 Cognachrome 一样，是不断设限的。其主要差别是：当对各颜色定义为 6 约束定界框体时，它使用的是 YUV 颜色空间，而不是 RGB 颜色空间。在 R，G，B 值将各颜色编码的同时，YUV 从亮度（或**照度**）度量中分离出颜色（或**色度**）度量。Y 表示图像的照度，而 U 和 V 共同获取色度。因此，以 YUV

空间表示的定界框体能达到比 RGB 空间所能做到的更大的稳定性(相对于照明变化)。

CMVision 颜色传感器达到 160×120 分辨率,并对各被检测物体,返回定界的框体和质心。CMVision 软件由参考文献[161]中 Gnu 的公用许可(证)免费提供。

对 CUMcam 软硬件和 Cognachrome 硬件系统,主要的特性瓶颈仍然是成像芯片的质量和可用的计算速度。随着在这些前沿技术取得重大进展,人们可以期待组装的视觉系统会展示特性指标的重大改进。

4.2 表示不确定性

在 4.1.2 节,我们提出了一个术语来描述传感器的指标特征。如前述那样,传感器是带有系统和随机性质误差的非理想装置。特别是随机误差不能被纠正,所以它们代表了传感器不确定性的原子水准。

但是,当你构建一个移动机器人时,你从许多传感器将信息组合,甚至重复使用同一传感器,那么随时间的消逝,你可能建立一个环境模型。我们如何从表征单个传感器的不确定性扩展到最终的机器人系统的不确定性呢?

通过提出与单个传感器[12]有关的随机误差统计的表示方法,我们开始讨论这个问题。用手中定量化的工具,可以提出和计算标准的高斯不确定性模型。最后,我们为了计算结论的不确定性提出了一个框架,而该结论是从称之为**误差传播律**的一组可定量的不确定性测量中导出的。

4.2.1 统计的表示

我们已经把**误差**定义为传感器测量值和真值之间的差。从统计

的观点看，我们希望表征传感器的误差，不只是对一个特定的测量，而是对任何的测量。让我们把感知问题形式化表示为估计问题。传感器已经获得了具有值 ρ_i 的 n 个测量集合。目标是给定这些测量，来表征真值 $E[X]$的估计：

$$E[X]=g(\rho_1,\rho_2,\cdots,\rho_n) \tag{4.50}$$

从这个概念，真值由一个随机(但为未知)变量 X 表示。我们用**概率密度函数**来表征 X 值的统计性质。

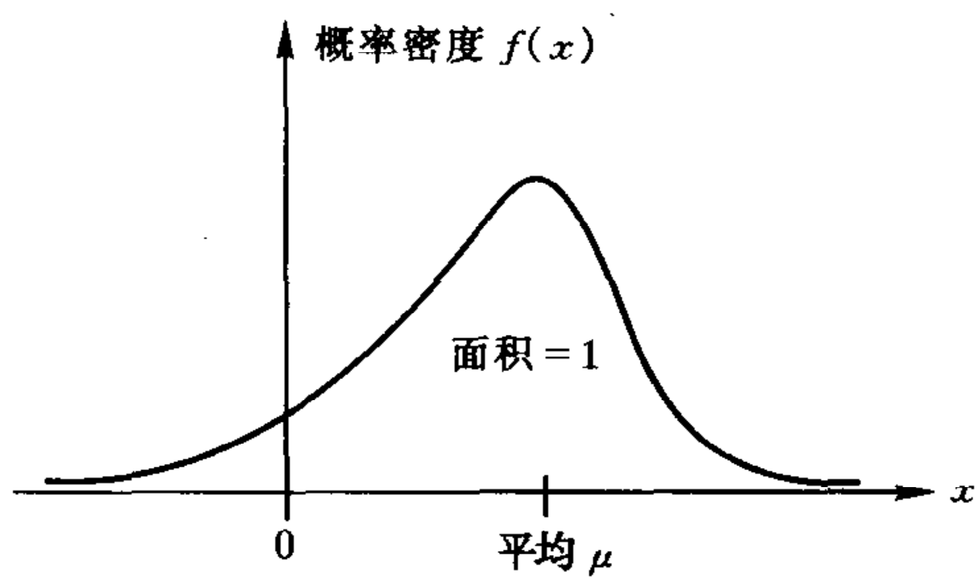


图 4.30 简单的概率密度函数，表示在两个方向上不对称跌落的单概率峰值(即单模)

在图 4.30 中，对 X 的各可能值 x，密度函数沿 y 轴确定了概率密度 $f(x)$。曲线下面的面积是 1，说明 X 的全部机会具有某个值：

$$\int_{-\infty}^{\infty} f(x)\mathrm{d}x = 1 \tag{4.51}$$

落在两个极限 a 和 b 之间的 X 值的概率，被计算为定积分：

$$P[a < X \leqslant b] = \int_a^b f(x)\mathrm{d}x \tag{4.52}$$

概率密度函数是表征 X 可能值的一个有用方法，因为它不仅获取了 X 的范围，而且获取了 X 不同值的相对的概率。利用 $f(x)$我们可以定量地定义均值、方差和标准偏差如下：

如果我们测量 X 无限次，并把全部的最终值平均起来，**均值** μ 就等效于期望值 $E[X]$：

$$\mu = E[X] = \int_{-\infty}^{\infty} xf(x)\mathrm{d}x \tag{4.53}$$

注意，上面计算 $E[X]$的方程等同于 x 的全部可能值加权平均。相反，**均方值**就是简单地将所有 x 值的平方加权平均：

$$E[X^2] = \int_{-\infty}^{\infty} x^2 f(x)\mathrm{d}x \tag{4.54}$$

X 可能值的“宽度”特征化是一个关键的统计测量，而且它要求首先定义方差 σ^2：

$$Var(X) = \sigma^2 = \int_{-\infty}^{\infty} (x-\mu)^2 f(x)\mathrm{d}x \tag{4.55}$$

最后，**标准偏差** σ 简单地是方差的平方根，且 σ^2 在我们表征单个传感器误差以及组合多个传感器数据所产生的模型误差中，起到重要作用。

4.2.1.1 随机变量的独立性

用上面提出的工具，我们常常计算具有多个随机变量的系统。例如，移动机器人的激光测距仪可用于测量机器人右边的特征位置；稍后，测量机器人左边的另一个特征。在现实环境中，各特征位置可以处理成随机变量 X_1 和 X_2。

如果一个变量的特定值不与另一特定值有关，则两个随机变量 X_1 和 X_2 是**独立的**。在这种情况下，我们可写出有关 X_1 和 X_2 统计行为的几个重要结论。第一，随机变量乘积的期望值（或均值）等于它们均值的乘积：

$$E[X_1X_2]=E[X_1]E[X_2] \tag{4.56}$$

第二，它们的和的方差等于它们方差的和：

$$Var(X_1+X_2)=Var(X_1)+Var(X_2) \tag{4.57}$$

在移动机器人中，我们常常假定随机变量是独立的，甚至当该假定不严格地为真时。如第 5 章所述，所做的简化使许多现有移动机器人的作图和导航算法得以成立。在 4.2.1.2 节中，进一步的简化则是围绕着一个特定的、在误差建模中比任何其他函数用得更多的概率密度函数：高斯分布。

4.2.1.2 高斯分布

当随机变量需要性能良好的误差模型，而该变量还未发现更恰当的误差模型时，就跨工程学科地使用高斯分布，它也被叫为**正态分布**。高斯具有很多特征，这使它在数学上优越于其他特定的概率密度函数。它对称于均值 μ。对大于或小于 μ 的值，无特殊的偏置。当无相反的信息时，这是有意义的。高斯分布也是单模的，有一个在 μ 处达到最大的单峰（对任何对称、单模分布，这是必需的）。该分布也有只渐近地趋向于零的尾部（当 x 趋向 $-\infty$ 和 $+\infty$ 时，$f(x)$ 的值）。这意味误差的所有结果都是可能的，虽然非常大的误差也许几乎不可能。在这意义上，高斯是保守的。最后，如在高斯概率密度函数方程所看到的那样，分布仅取决于两个参数：

$$f(x)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4.58}$$

高斯的基本形式决定于该方程的结构，所以，要完全地指定一个特殊的高斯，仅需要两个参数：均值 μ 和它的标准偏差 σ。图 4.31 表示了 $\mu=0$ 和 $\sigma=1$ 的高斯函数。

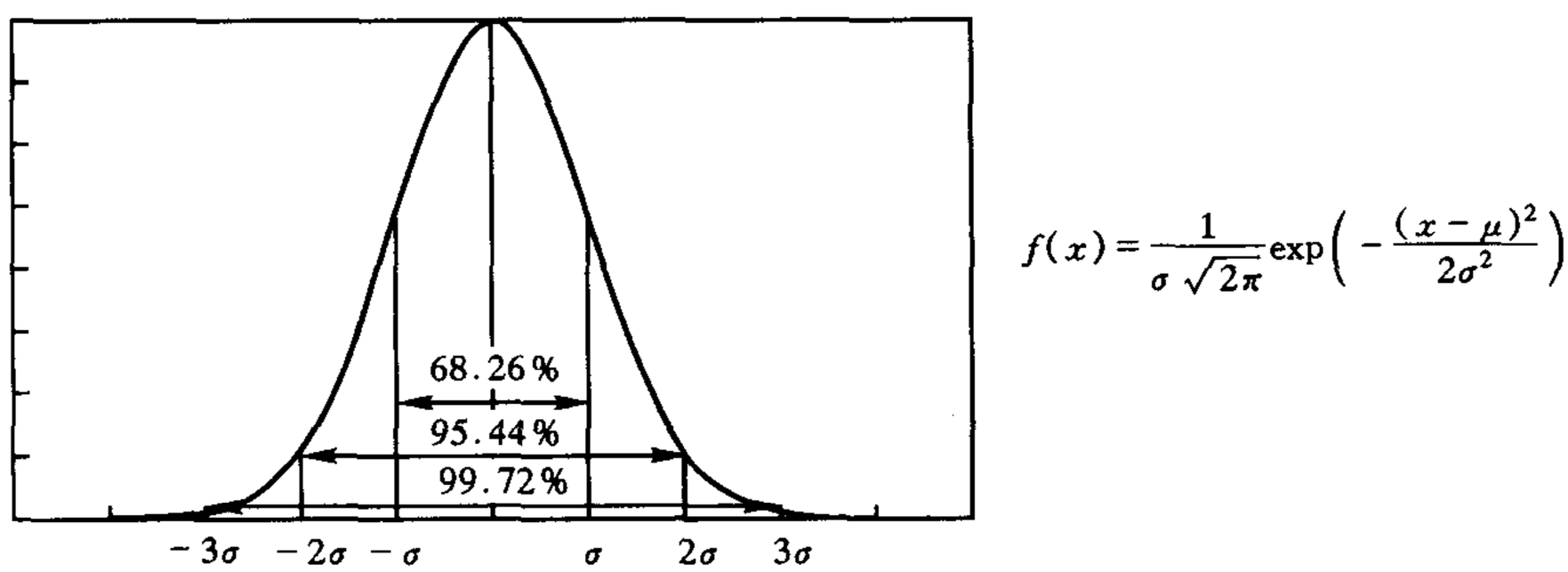


图 4.31 $\mu=0$ 和 $\sigma=1$ 的高斯函数。我们应该把此当作参考高斯。2σ 的值常常当作信号的质量；值的 95.44%落在 $\pm 2\sigma$ 范围内

假定一个随机变量 X 被建模成一个高斯，如何确定 X 的值是在标准偏差 μ 的范围内的机会呢？实际上，这需要对高斯函数 $f(x)$ 的积分，计算曲线下面部分的面积：

$$面积 = \int_{-\sigma}^{\sigma} f(x)\mathrm{d}x \tag{4.59}$$

遗憾的是，对方程(4.59)积分不存在闭合解。因此公用的技术是利用高斯的**累积概率表**。利用这个表，我们可以对 X 的不同值域计算概率：

$$P[\mu-\sigma<X\leqslant\mu+\sigma]=0.68;$$
$$P[\mu-2\sigma<X\leqslant\mu+2\sigma]=0.95;$$
$$P[\mu-3\sigma<X\leqslant\mu+3\sigma]=0.997。$$

例如，对 X 来说，其值的 95%落在它均值的两个标准偏差之内。这适用于**任何**高斯分布。从以上的进行步骤看，这是清楚的。在高斯假定下，一旦界定放松到 3σ，则值的整个部分(从而概率)都被包括在内。

4.2.2 误差传播：对不确定的测量进行组合

以上的概率机制可以用来描述与欲测的现实环境值的单个传感器相关联的误差。但在移动机器人学中，人们往往利用全是不确定的一系列测量，以提取单个环境的度量。例如可以融合单点的一系列不确定性测量，以提取环境中一条直线(如走廊的墙)的位置(图 4.36)。

考虑图 4.32 的系统，图中 X_i 是具有已知概率分布的 n 个输入

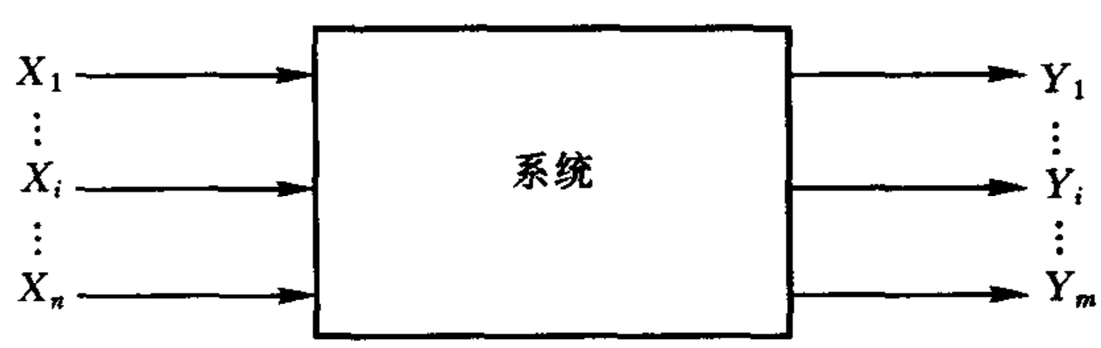


图 4.32 在具有 n 个输入和 m 个输出的多输入多输出系统中误差的传播

信号，Y_i 是 m 个输出。有趣的问题是：在已知函数 f_i 的情况下，如果输出依赖于输入信号，则有关输出信号 Y_i 的概率分布是什么？作为一个例子，图 4.33 描述了这个误差传播的 1D 式样。

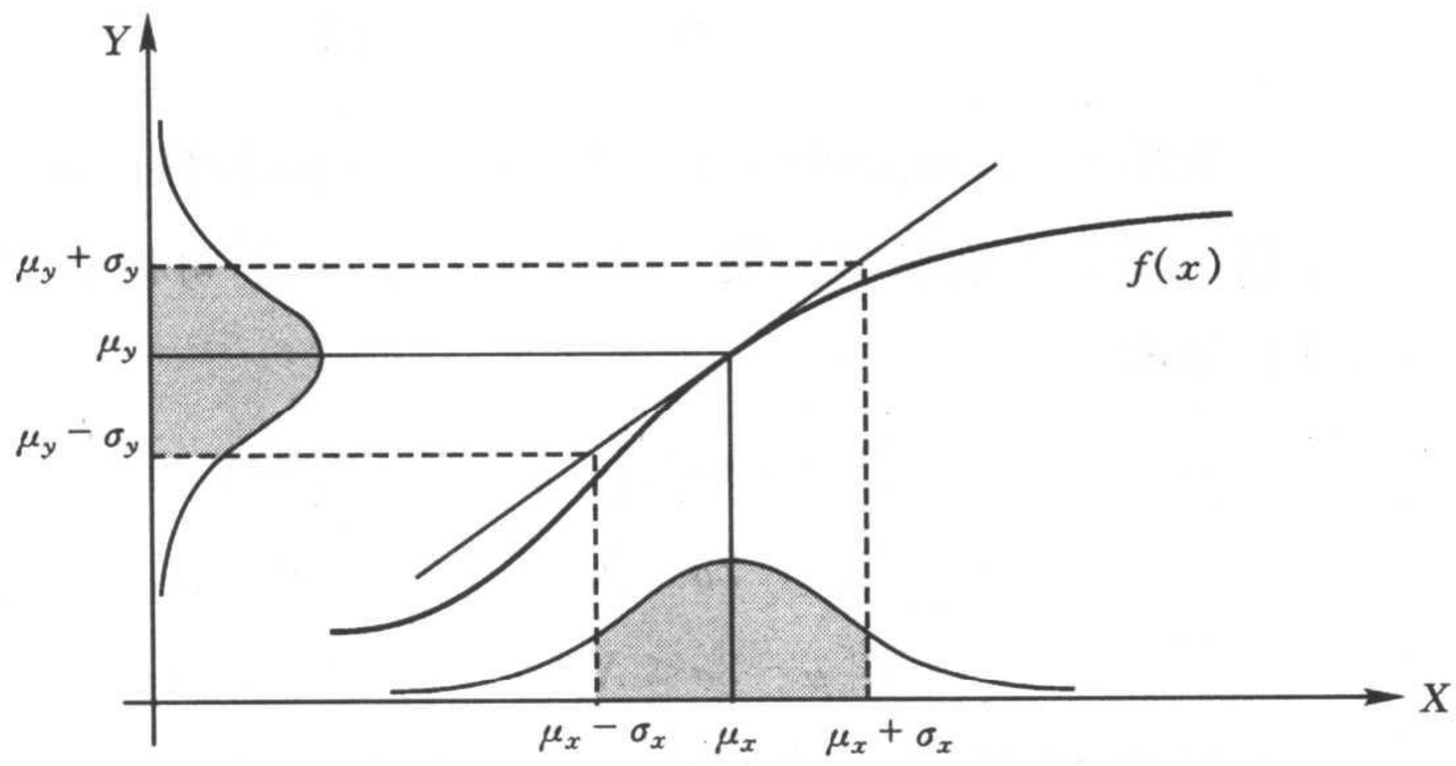


图 4.33 非线性传播问题的一维情况

利用 f_i 一阶泰勒展开式我们可以得到通用解。输出的协方差矩阵 $\boldsymbol{C}_Y$ 由误差传播律给出：

$$\boldsymbol{C}_Y = F_X C_X F_X^T \tag{4.60}$$

式中

$\boldsymbol{C}_X$ = 表示输入不确定性的协方差矩阵；

$\boldsymbol{C}_Y$ = 表示输出转播不确定性的协方差矩阵；

$\boldsymbol{F}_x$ 是**雅可比**矩阵，定义为：

$$\boldsymbol{F}_X = \nabla f = [\nabla_X \cdot f(X)^T]^T = \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix} \left[\frac{\partial}{\partial X_1} \cdots \frac{\partial}{\partial X_n} \right]$$

$$= \begin{bmatrix} \frac{\partial f_1}{\partial X_1} & \cdots & \frac{\partial f_1}{\partial X_n} \\ \vdots & \cdots & \vdots \\ \frac{\partial f_m}{\partial X_1} & \cdots & \frac{\partial f_m}{\partial X_n} \end{bmatrix} \tag{4.61}$$

这也是 $f(X)$ 梯度的转置。

这里，我们不进行详细的推导，但我们将利用方程(4.60)求解 4.3.1.1 节中一个实例问题。

4.3 特征提取

一个自主移动机器人必须能够用它的传感器进行测量，然后利用这些测量信号确定它与环境的关系。如在前面章节所示，有种类繁多的感知技术可以利用。但我们已提供的每一种传感器都不是理想的：测量经常有误差，以及与它们相关联的不确定性。但是，尽管测量有不确定性，我们也必须以一种方法使用传感器的输入，使机器人能与它的环境成功地进行交互。

这里存在两种用不确定的传感器引导机器人的策略。一种策略是用各传感器的测量作为原始的和单个的值。这种原始传感器的值，例如可以被直接联系到机器人的行为，以使机器人的动作是它传感器输入的函数；另一种选择是传感器的原始值可以被用于更新即时模型，这样，被激发的机器人的动作成为模型的函数，而不是单个传感器测量的函数。

第二种策略是首先从一个或多个传感器数据中提取信息，产生较高级的**感觉**，然后用它通知机器人的模型，也可直接通知机器人的动作。我们称这个过程为**特征提取**。接下来我们讨论在感知解释流程(图 4.34)中一个可选的步骤。

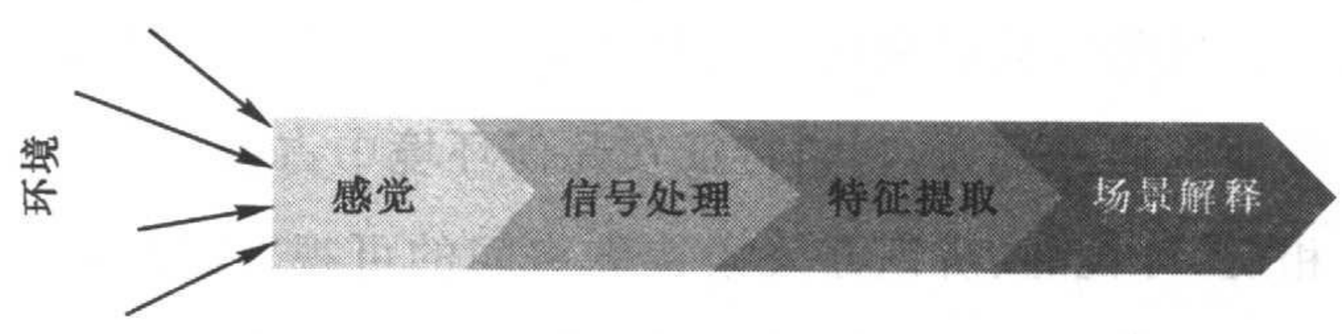


图 4.34 感知流程：从传感器模型到知识模型

在实际中，移动机器人并不需要对每一个动作使用特征提取和场景解释，代之以根据各特定机能，机器人在可变的范围内解释传感器。例如，为了在面临即时障碍情况下保证紧急停止，机器人可直接

使用原始的正面距离数据停止它的驱动电机。对局部避障来说，在占用栅格的模型中可以组合原始测距传感器的测量，使之在几米远避开避障。而对作图和精确的导航来说，测距传感器的值，甚至是视觉传感器的测量，都可以经历完整的感知流程。它以特征提取为依据，紧随着场景解释使单个传感器的不确定性对机器人制图和导航技巧的鲁棒性影响最小。因此出现的模式是：随着人们参与更复杂、长期的感知任务，感知流程的特征提取和场景解释变得十分重要。

特征定义 特征是环境中可认识的元素结构。它们通常可从测量和数学描述中提取出来。良好的特征常常是可感知的，且易于从环境中检测。我们要区分**低级特征**(几何的基本要素)，像直线、圆或多边形，和**高级特征**(物体)，诸如边缘、门、桌子或垃圾桶之间的区别。在一个极端，原始传感器提供大容量的数据，但各个体数据量的特殊性少。使用原始数据有潜在的优点，即充分利用了信息的每一位，因而有高的信息保守性。低级特征是原始数据的抽象，这样，提供较小的数据容量，同时又增加各特征的特殊性。当人们整合低级特征时，希望特征过滤掉差的或无用的数据。当然，特征提取过程的结果也可能失去某些有用的信息。高级特征从原始数据中提供最大的抽象性，由此尽可能地减少数据的容量，同时又提供了独特的最终特征。再次需要提醒的是，抽象过程有过滤掉重要信息的危险，从而潜在地降低了数据的实用性。

虽然特征必须具有某些空间的局部性，但它们的几何区域可广泛地延伸。例如，拐角特征在几何环境中占据一个特定的坐标位置。相反，可视的、辨识办公楼特殊房间的可视“指印”适用于整个房间，但在空间上限于在特殊房间中拥有一个位置。

在移动机器人学建立环境模型中，特征起着特殊重要的作用。它们能够更精密和鲁棒地描述环境，在作地图和定位期间帮助移动机器人。在设计一个移动机器人时，关键的决策要反复考虑选择合适的特征以供机器人使用。许多因素对决策是很关键的。

目标环境 为了使几何特征有用，目标的几何特性在实际环境中必须易于检测。例如，在办公楼环境中，由于存在大量平直的墙线段，直线特征极其有用；而同样的特征在导航火星时，实际上却是无用的。

可用的传感器 显然，机器人的特定的传感器和传感器的不确定性影响着不同特征的适应性。由于激光扫描器高质量的角度和深度分辨率，装备上激光测距仪的机器人完全有条件使用几何上详细的特征，诸如拐角特征。相反，装有声纳的机器人可能没有拐角特征提取的合适工具。

计算的能力 基于视觉的特征提取可以招致巨大的计算成本。特别是在在机器人中，视觉传感器的处理由机器人的主处理器执行。

环境表示 特征提取是朝向场景解释的一个重要步骤。由此看来，所提取的特征必须提供与环境模型所用的表示方法相一致的信息。例如，非几何的基于视觉的特征在纯几何环境模型中很少有价值，但在环境的拓扑模型中有很大价值。图 4.35 表示了用于办公楼

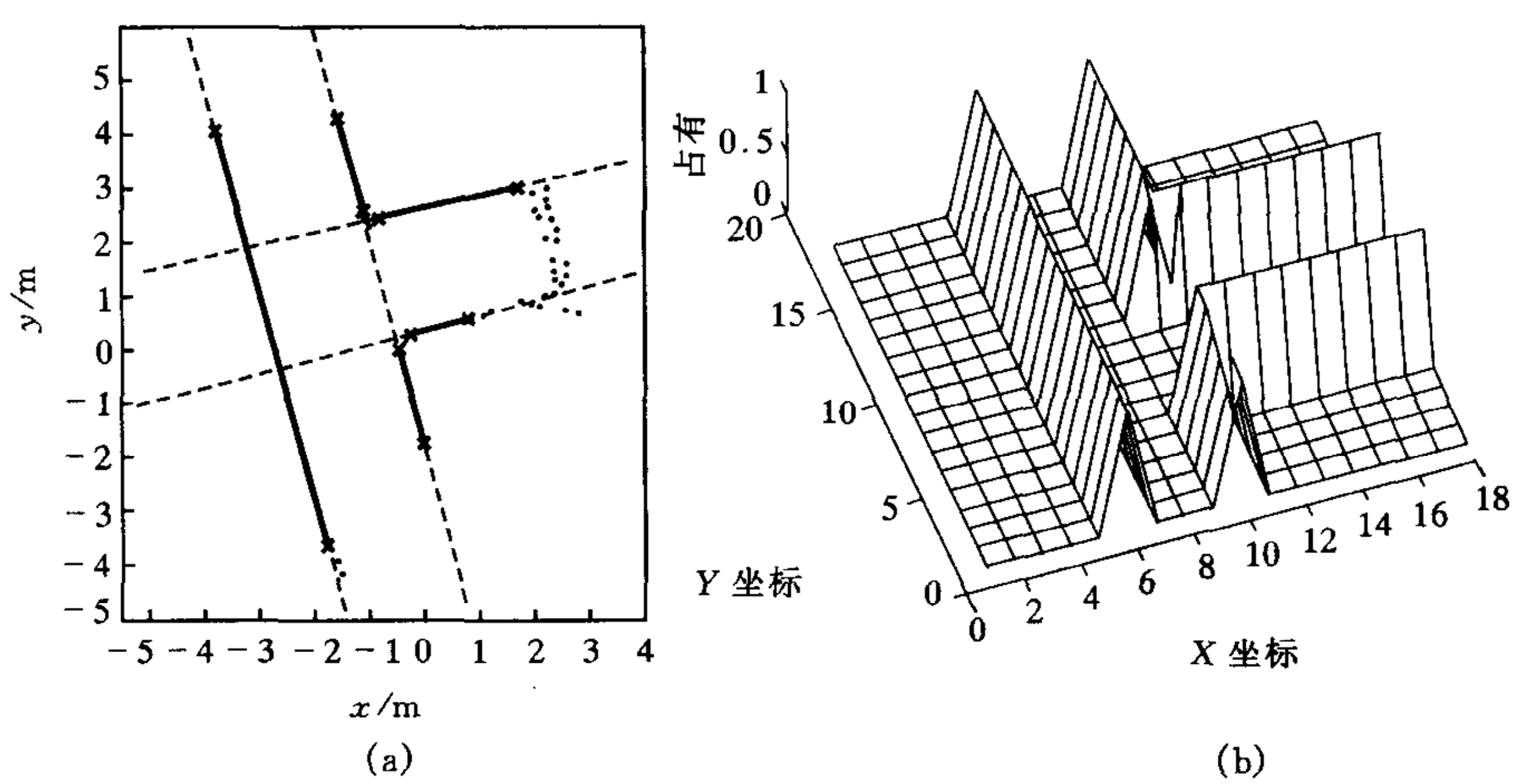


图 4.35 环境表示方法和建模：(a) 基于特征(连续度量)；(b) 占有格(离散度量)。图片由 Sjur Vestli 提供

走廊建模任务的两种不同的表示方法。各个方法各有缺点和优点，但是直线和拐角特征的提取与左边表示方法有更多的关系。为详细

观察地图的表示方法和它们相关的折衷，请参考第5章5.5节。

在下面两节，根据移动机器人学两个最普遍的感知模式：距离感知和基于表象的视觉感知，提出特定的特征提取技术。

4.3.1 基于距离数据的特征提取(激光、超声和基于视觉测距)

今天大多数从测距传感器提取的特征都是几何基本元素，如线段或圆。其主要的理由是，对大多数其他几何元素来说，特征的参数描述太复杂，并且不存在闭式解。这里我们详细描述直线提取，展示以上提出的不确定性模型如何可以被应用于多传感器组合的问题。此后，我们简要地提出室内机器人另外一个很成功的拐角特征，并说明这些特征如何可以被整合成一个单独的表示。

4.3.1.1 直线提取

几何特征提取通常是将被测的传感器数据与期望特征的预定描述或模板进行比较和匹配的过程。通常，传感器测量的数目超过被估计的特征参数的数目，系统是超定的。因为传感器的测量都存在某些误差，故没有理想的一致解，而是一个优化问题。例如，人们可以提取特征，使之与全部传感器所用的测量值之间的差异最小(最小平方估计)。

在本节，我们对从不确定的传感器测量集合中提取直线特征的问题，提出一个基于优化的解决方案。比下述更为详细的描述请参考文献[14，pp. 15和221]。

从不确定的距离传感器数据中提取概率直线 我们的目标，如图4.36所示，是根据传感器的测量集合提取一条直线。这里存在着与各个带噪声的距离传感器测量有关的不确定性，所以不存在通过集合的单条直线，而是给定某些优化的准则，我们希望选择最佳可能的匹配。

从形式上看，假定机器人传感器处在极坐标中，产生 n 个距离测

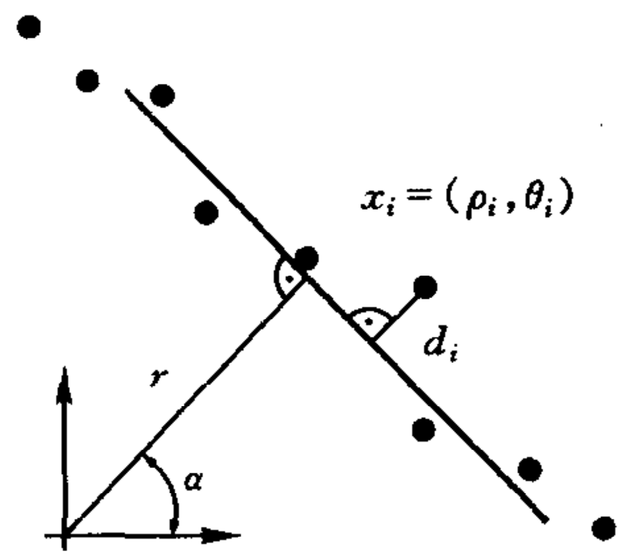


图 4.36 在最小平方意义上估计直线。模型参数 r(垂直长度)和 α(与横坐标的角度)唯一地描述直线

量点 $x_i=(\rho_i,\theta_i)$。我们知道存在与各测量有关的不确定性。所以，我们可以用两个随机变量 $X_i=(P_i,Q_i)$ 对各测量进行建模。在分析中，我们假定相对于实际值 P 和 Q 的不确定性是独立的。根据方程(4.56)，我们可在形式上阐述为：

$$E[P_i \cdot P_j]=E[P_i]E[P_j] \quad \forall i,j=1,\cdots,n \tag{4.62}$$

$$E[Q_i \cdot Q_j]=E[Q_i]E[Q_j] \quad \forall i,j=1,\cdots,n \tag{4.63}$$

$$E[P_i \cdot Q_j]=E[P_i]E[Q_j] \quad \forall i,j=1,\cdots,n \tag{4.64}$$

而且，我们假定各个随机变量受高斯概率密度曲线的约束，真值为均值，具有某些特定的方差：

$$P_i \sim N(\rho_i,\sigma_{\rho_i}^2) \tag{4.65}$$

$$Q_i \sim N(\theta_i,\sigma_{\theta_i}^2) \tag{4.66}$$

给定某些测量点 (ρ,θ)，我们可以计算相应的欧氏坐标为 $x=\rho\cos\theta$ 和 $y=\rho\sin\theta$。如果没有误差，我们要寻求一条直线，使全部测量点都在那条直线上：

$$\rho\cos\theta\cos\alpha+\rho\sin\theta\sin\alpha-r=\rho\cos(\theta-\alpha)-r=0 \tag{4.67}$$

当然，因为存在测量误差，所以上式不为零。当它非零时，特别地，根据点和直线之间最小的正交距离，这就是测量点 (ρ,θ) 和直线之间误差的度量。了解如何对应该最小化的误差进行测量常常是重要的。例如，许多直线提取技术并不使正交的点-线距离最小，而是使点线之间并行于 y 轴的距离最小。在参考文献[17]中可以看到

各种优化准则的很好说明。在该文献里，提出了几个使代数和几何距离最小的拟合圆和椭圆的算法。

对各特定的(ρ_i,θ_i)，我们可以写出(ρ_i,θ_i)和直线之间的正交距离d_i为

$$\rho_i\cos(\theta_i-\alpha)-r=d_i \tag{4.68}$$

如果我们考虑各测量有相同的不确定性，对全部的测量点我们可以将所有误差的平方加在一起，将直线和全部测量之间的整个拟合进行量化：

$$S=\sum_i d_i^2=\sum_i(\rho_i\cos(\theta_i-\alpha)-r)^2 \tag{4.69}$$

我们的目标是，在选择直线参数(α,r)时使S最小。通过求解非线性方程组，我们可以做到：

$$\frac{\partial S}{\partial\alpha}=0 \quad 和 \quad \frac{\partial S}{\partial r}=0 \tag{4.70}$$

上面的方程被认为是**不加权的最小平方解**，因为在测量点之间没有造成差别。在现实中，当记录测量时，根据机器人和环境的几何特性，各传感器的测量可能有它自己独特的不确定性。例如，关于基于视觉的立体测距，我们知道它的不确定性和方差随机器人和物体之间距离的平方而增加。为了使用特殊传感器测量的关于距离ρ_i不确定性建模的方差σ_i^2，我们用以下方程对各测量计算一个单独的权值：

$$w_i=1/\sigma_i^2 \ ^* \tag{4.71}$$

由此，方程(4.69)变为

$$S=\sum w_i d_i^2=\sum w_i(\rho_i\cos(\theta_i-\alpha)-r)^2 \tag{4.72}$$

可以证明，在加权最小平方意义上**，方程(4.70)的解为：

---

* 当σ_i给定(也许某些附加信息)，确定合适的权值问题一般是复杂的，且超出本书范围，为慎重处理，请参阅参考文献[9]。

** 这里我们遵循参考文献[14]的概念，并区分加权最小平方问题，如果C_X是对角的(输入误差相互独立)和广义最小平方问题，如果C_X是非对角的。

$$\alpha = \frac{1}{2}\arctan\left(\frac{\sum w_i\rho_i^2\sin 2\theta_i - \frac{2}{\Sigma_{w_i}}\sum\sum w_i w_j \rho_i\rho_j\cos\theta_i\sin\theta_j}{\sum w_i\rho_i^2\cos 2\theta_i - \frac{1}{\Sigma_{w_i}}\sum\sum w_i w_j \rho_i\rho_j\cos(\theta_i+\theta_j)}\right) \tag{4.73}$$

$$r = \frac{\sum w_i\rho_i\cos(\theta_i-\alpha)}{\sum w_i} \tag{4.74}$$

实际上，方程(4.73)使用了四象限的反正切(arctan2)。*

让我们用一个具体例子说明方程(4.73)和(4.74)。用一个装在移动机器人上的激光距离传感器，已获得了表 4.2 中 17 个测量值

表 4.2 测量值

传感器的指示角 θ_i/(°)	距离 ρ_i/m
0	0.5197
5	0.4404
10	0.4850
15	0.4222
20	0.4132
25	0.4371
30	0.3912
35	0.3949
40	0.3919
45	0.4276
50	0.4075
55	0.3956
60	0.4053
65	0.4752
70	0.5032
75	0.5273
80	0.4879

* (arctan2) 计算 $\tan(x/y)^{-1}$，但使用 x 和 y 二者的符号以确定最终角度所在的象限。例如，arctan(−2，−2) =− 135°，而 arctan2(2,2) =− 45°。如用单自变量反正切函数，差别就会被丢失。

(ρ_i, θ_i)。我们假定在测量过程中，所有测量的不确定性是相等、不相关的，且机器人是静态的。

直接用上面的求解方程，得到由 $\alpha=37.36$ 和 $r=0.4$ 的直线。该直线代表在最小平方意义上最好的拟合，并表示于图 4.37。

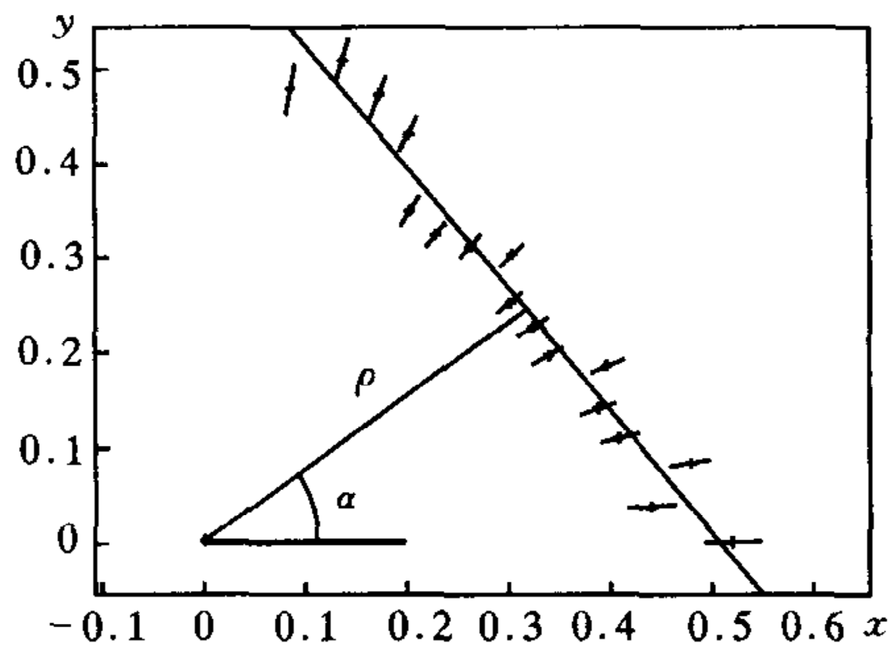


图 4.37 从激光距离测量(+)中提取直线。在各测量点的短线表示测量的不确定性 σ，它正比于测量距离的平方

在直线提取期间不确定性的传播 让我们回到 4.2.3 节的主题，我们想了解特定传感器测量的不确定性如何传播以支配所提取直线的不确定性。换句话说，在方程(4.73)和(4.74)中 ρ_i 和 θ_i 如何传播以影响 α 和 r 的不确定性？

这要求直接应用方程(4.60)，式中 A 和 R 分别表示随机输出变量 α 和 r。目标是给定 $2n\times 2n$ 输入协方差矩阵

$$C_X=\begin{bmatrix} C_P & \mathbf{0} \\ \mathbf{0} & C_Q \end{bmatrix}=\begin{bmatrix} \mathrm{diag}(\sigma_{\rho_i}^2) & \mathbf{0} \\ \mathbf{0} & \mathrm{diag}(\sigma_{\theta_i}^2) \end{bmatrix} \tag{4.76}$$

和系统关系式(方程(4.73)和(4.74))，推导 2×2 的输出协方差矩阵

$$C_{AR}=\begin{bmatrix} \sigma_A^2 & \sigma_{AR} \\ \sigma_{AR} & \sigma_R^2 \end{bmatrix}, \tag{4.75}$$

然后通过计算雅可比

$$F_{PQ}=\begin{bmatrix}\frac{\partial\alpha}{\partial P_1} & \frac{\partial\alpha}{\partial P_2} & \cdots & \frac{\partial\alpha}{\partial P_n} & \frac{\partial\alpha}{\partial Q_1} & \frac{\partial\alpha}{\partial Q_2} & \cdots & \frac{\partial\alpha}{\partial Q_n}\\ \frac{\partial r}{\partial P_1} & \frac{\partial r}{\partial P_2} & \cdots & \frac{\partial r}{\partial P_n} & \frac{\partial r}{\partial Q_1} & \frac{\partial r}{\partial Q_2} & \cdots & \frac{\partial r}{\partial Q_n}\end{bmatrix} \tag{4.77}$$

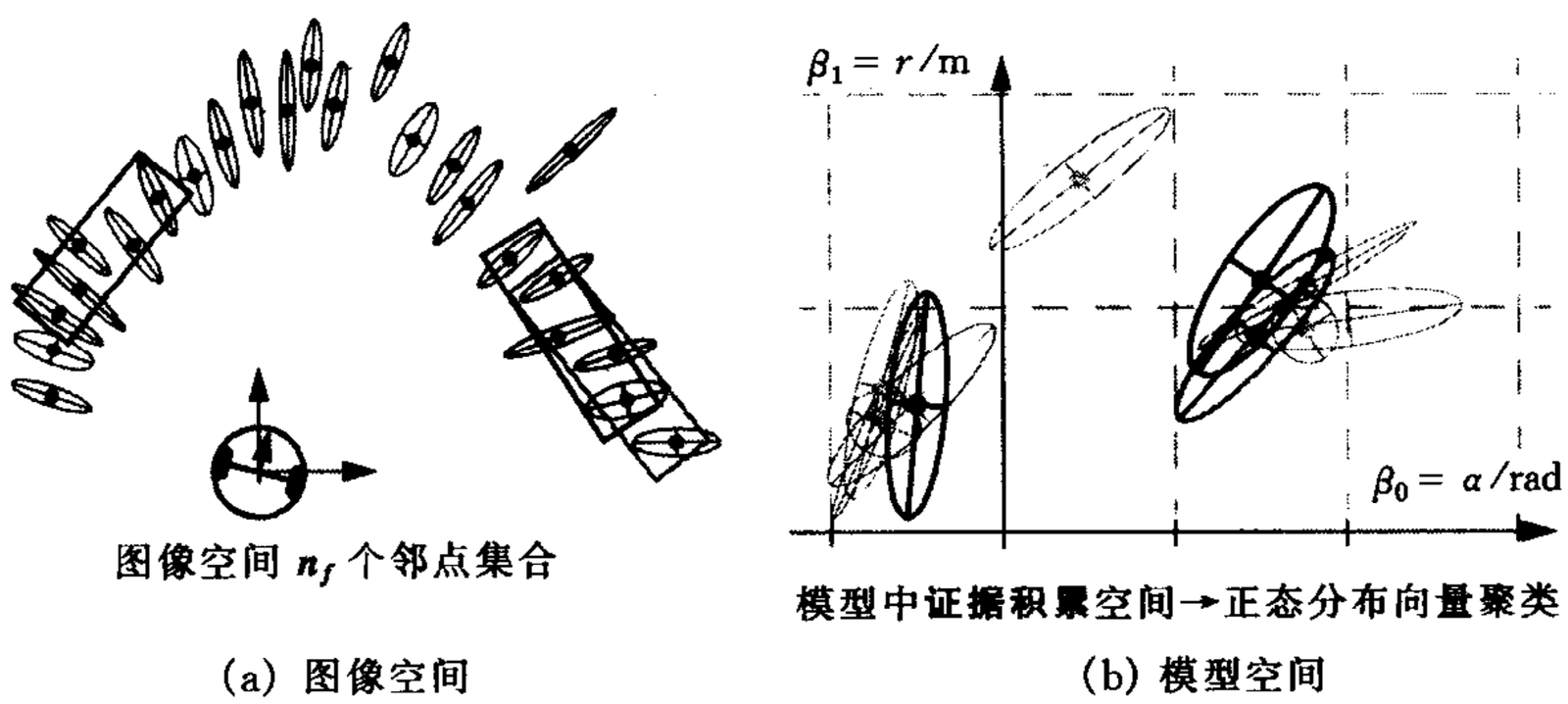
我们可以示例不确定性传播方程(4.60),得到 C_{AR}:

$$C_{AR}=F_{PQ}C_X F_{PQ}^T \tag{4.78}$$

由此,我们根据测量点的概率,已经计算出所提取直线(α,r)的概率 C_{AR}。关于该方法的更详细资料,请参考文献[6,37]。

4.3.1.2 直线提取的分割

前一节描述了给定距离测量的集合如何提取直线特征。遗憾的是,特征的提取过程比特征复杂得多。移动机器人确实获取了距离测量的集合,但一般距离测量不是一条直线的全部,而只有距离测量的某些部分在直线提取中起作用。更进一步地说,在测量集合里也许表示了一个以上的直线特征。这个更现实的场景表示于图 4.38。



(a) 图像空间　　(b) 模型空间

图 4.38　聚类:寻找公共直线的相邻段[37]

把测量集合划分成子集,使它可以依次地被解释,这过程称为**分割**,它既是基于距离也是基于视觉感知的重要方面。对传感器的输入分割,一般存在不同的成套技术。该一般问题超出了本书的范围,关于分割算法的更详细资料,可参考文献[91,131]。

例如,一种分割技术是合并,或**自底向上**的技术。在该技术中,辨识较小的特征,然后根据决策准则合并在一起,提取目标特征。假

定，通过合并解决了图 4.38 中的问题。首先，人们根据距离测量的相邻群，产生多个直线段。第二步是辨识属于同一个所提取的直线特征其概率高的线段。在模型空间中，两个线段* $x_1=[\alpha_1, r_1]$和 $x_2=[\alpha_2, r_2]$接近度的最简单度量由欧氏距给定：

$$(x_1-x_2)^T(x_1-x_2)=(\alpha_1-\alpha_2)^2+(r_1-r_2)^2 \tag{4.79}$$

现在，对同一直线作贡献的所有直线段 x_j 作选择。可以按照下式按基于阈值的方式进行：

$$(x_j-\bar{x})^T(x_j-\bar{x})\leqslant d_m \tag{4.80}$$

式中 d_m 是阈值，$\bar{x}$ 表示参考直线(由一个模型、直线群的平均等所产生)。

但是方程(4.80)的方法没有考虑到以下的事实：对各测量，并由此对各线段，存在不确定性的测量。通过选择由它们的协方差矩阵 C_j 加权的线段，我们可对该方程进行改进：

$$(x_j-\bar{x})^T(C_j-\bar{C})(x_j-\bar{x})\leqslant d_m \tag{4.81}$$

方程(4.81)的距离测量经过明显地考虑不确定性，更有效地识别了模型空间中不确定点的距离。

4.3.1.3 距离直方图特征

直方图是组合图像特征元素的一个简单方法。如图 4.39 所示，角度直方图绘制了由两个相邻距离测量所提取的直线的统计。首先用距离扫描仪取得一个房间 360°的扫描，且最终的“采样数”被记录在地图中。然后算法测量任何两个相邻采样数之间的相对角度(图 4.39(b))。在读数中补偿噪声(相邻采样间位置不准确引起)之后，就可以制作图 4.39(c)所示的角度直方图。与主墙统一的方向在角度直方图中，清楚地可视为峰值。峰值检测只获得两个主峰：一个主峰对应一对平行墙。该算法对于墙中缺口，诸如门、窗或者甚至是嵌在墙内的橱柜，鲁棒性都非常好。

---

* 直线在极坐标系表示。

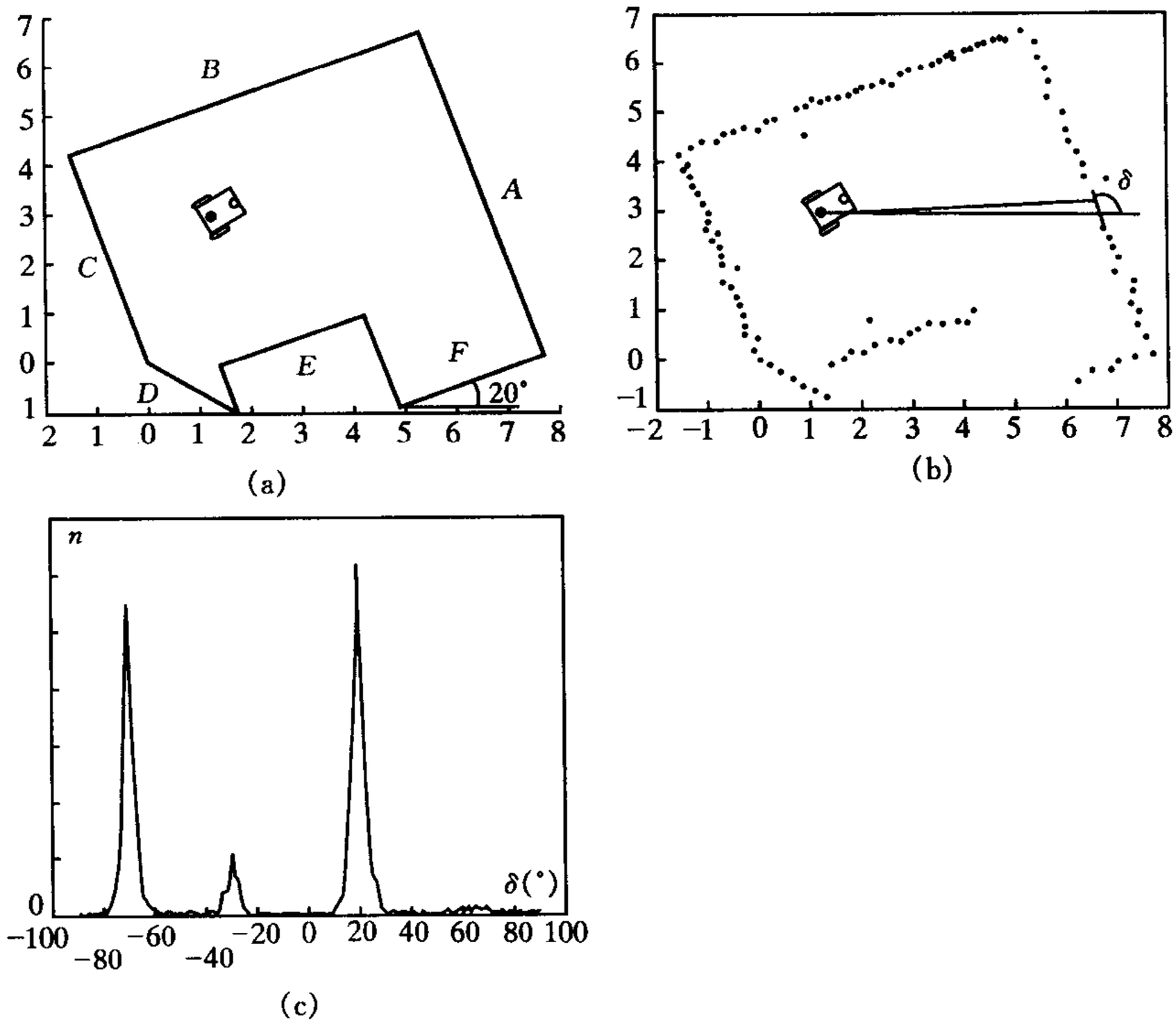


图 4.39 角度直方图

4.3.1.4 提取其他几何特征

直线特征对运行在人造环境中的移动机器人特别有价值。例如，大楼的墙和走廊的墙通常都是直的。一般，移动机器人同时使用多个特征，组成最适合于它运行环境的**特征集合**。对室内移动机器人，直线特征肯定是最优特征集合的一个成员。

另外，贯穿室内人造的环境总是出现其他的几何特征。**拐角特征**定义为具有方向的点特征。**台阶间断**定义为垂直于走廊行进方向的台阶变化，由它们的形式（凹或凸）和台阶大小来表征。

由此，标准的分割问题不像从传感器读数到线段确定映射那样简单，而是一个过程。在该过程里，根据可用的传感器数据提取不同类型的特征。连同锯齿形缺口特征（即台阶断续性）和门口在内，图

4.40表示了室内走廊的环境。

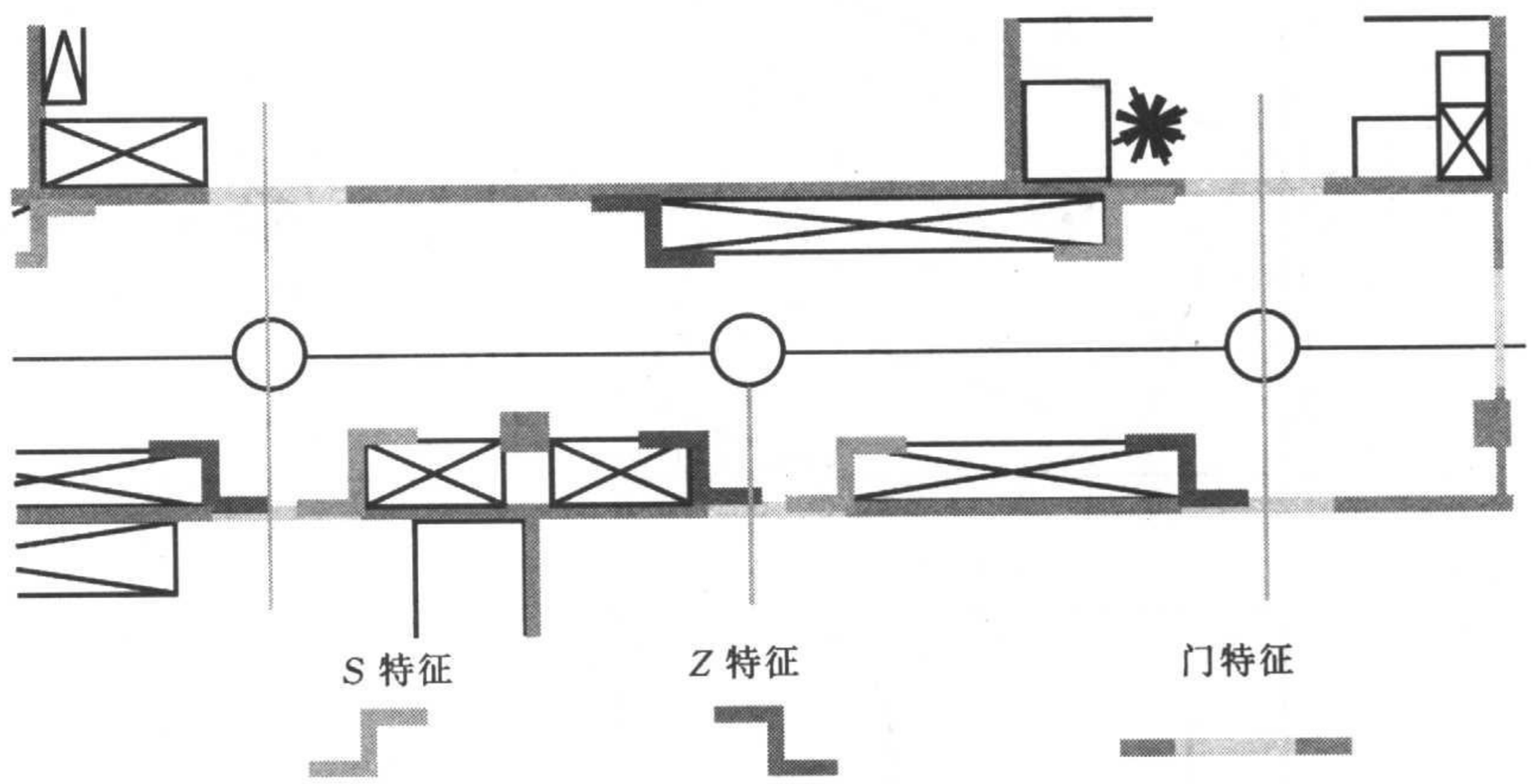


图 4.40 在单个走廊里多个几何特征，包括门口和走廊宽度中的间断

需要注意的是，不同的特征类型可为移动机器人定位提供定量化的不同信息。例如，直线特征的二度信息：角度和距离。但台阶特征提供了2D的相对位置和角度信息。

有用的几何特征集合基本上是无限制的，随着传感器特性指标的改善，我们只能期望在特征提取的水平上有更大的成功。例如，对上述直线特征有意义的改进，关系到成功的基于视觉系统的进展。因为立体视觉提供全3D的距离测量的集合。除了从最终数据提取直线特征外，我们可以提取平面特征。平面特征在人造环境中，由于我们室内有平坦的墙、地板和天花板，所以是有价值的。因此，对用于作图和定位的移动机器人，平面特征有希望成为另一种有高度信息的特征。

4.3.2 基于可视表象的特征提取

正如我们以前已指出的那样，视觉解释的完全解决是一个具有极大挑战性的问题。在过去的几十年里，为理解基于2D图像的场景，已投入了巨大的研究力量来创新算法，且研究工作已缓慢地产生

了丰硕的成果。当然，要覆盖计算机视觉和图像处理领域，则超出本书的范畴。为探索这些学科，请参考文献[18,29,159]。在图 4.41 中，我们可以看到关于最普遍方法某些方面的综述。

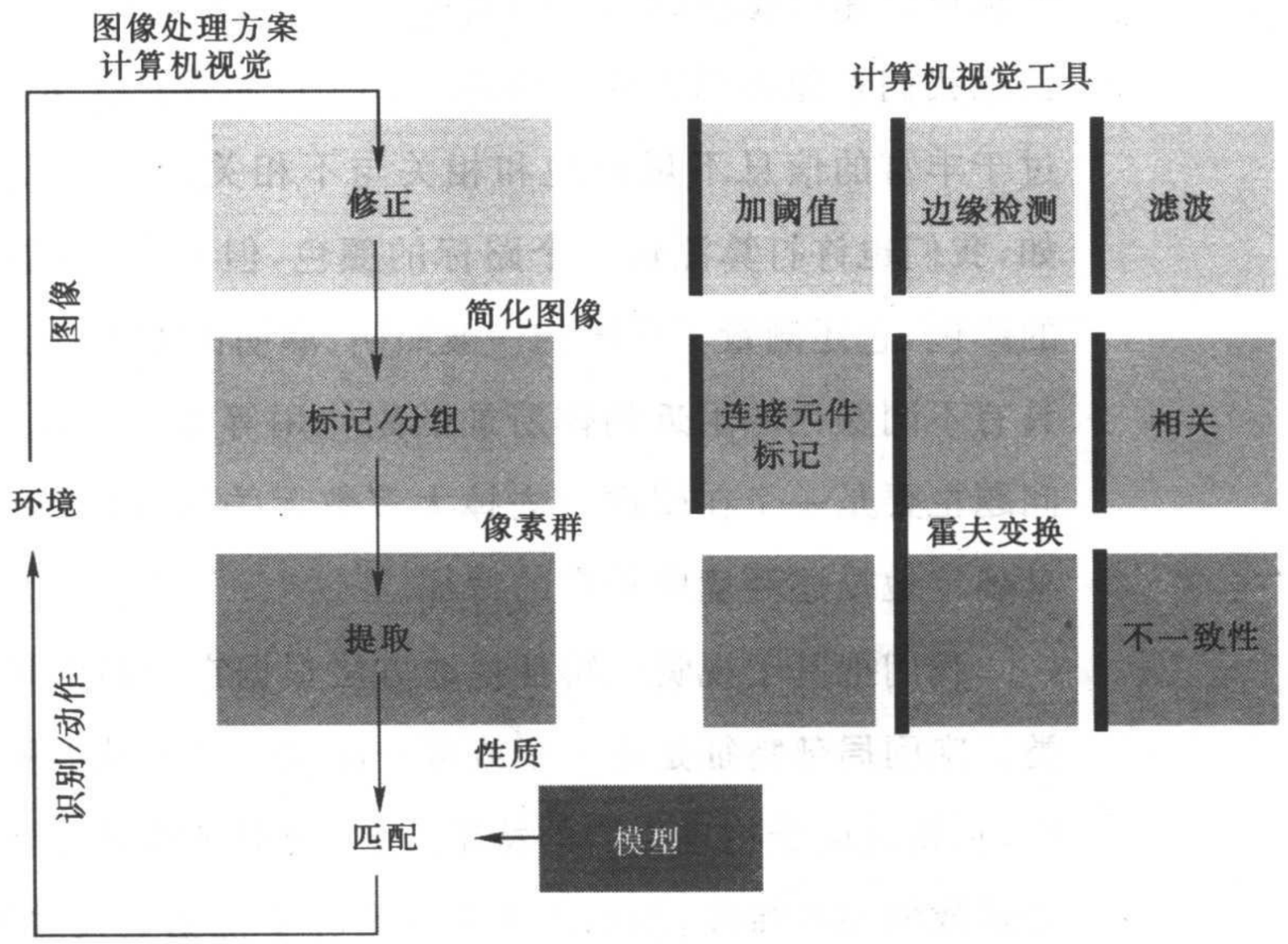


图 4.41 计算机视觉的方案和工具，也可参考[18]

在 4.1.8 节，我们已经看到了基于视觉的测距和商业上移动机器人可用的颜色跟踪传感器。这些特定的视觉应用，基本上为商用的解决方案作了证明，因为在两种情况下，比较好地对准了难点，且最终针对特定问题的算法是直观明了的。但是图像包含了比隐含的深度信息和色斑多得多的内容。我们希望解决从图像中提取大量特征类型的更一般问题。

本节提出某些基于表象的特征提取技术，它处处与移动机器人学相关。为了使基于视觉的特征提取技术具有移动机器人的相关性，必须满足两个关键的要求。第一，方法必须实时运行，移动机器人经过它们的环境而运动，所以处理不能是简单地离线操作。第二，方法必须对实验室以外的现实环境条件是鲁棒的。这意味着谨慎控

制的照明假设和仔细着色的物体都是不能接受的要求。

在下面的描述中要记住，基于视觉的解释基本上是关于对**缩减信息**的挑战。声纳单元大概产生每秒50位的信息。相反，CCD摄像机可以输出每秒240,000,000位信息！声纳产生少量信息，从这些信息我们希望得到主要的结论；但CCD芯片产生太多信息，而且该过于丰富的信息不规则地和相关与不相关的信息混杂在一起。例如，我们也许打算测量一个路标的颜色，但CCD不是简单地报告它的颜色，它还测量了环境的一般照明、照明的方向、光学造成的散焦、具有不同颜色的附近物体所加的副作用等等。所以，视觉特征提取问题主要是一个在图像中去掉大多数无关的信息，使得留下来的信息确定地描述环境中特定的特征。

我们把基于视觉的特征提取方法根据它们的空间延伸划分成两类。**空间局部特征**是在一个或多个图像的子区域中所发现的那些特征，它相对应于物理世界的特定位置；**全图像特征**是全图像或图像集合函数的那些特征，相对应于物理世界中视觉上大的连接区。

在继续讨论之前，重要的是要注意到，所有基于视觉的传感器提供了带有如此大量噪声的图像，使得在启用任何特征提取算法之前第一步总是包含“清理”图像。所以，我们首先描述初始图像滤波或预处理的过程。

图像预处理 许多图像处理算法使用图像强度的二次导数。确实，我们在4.1.8.2节研究的用于立体测距的高斯拉普拉斯方法就是这种实例。由于这种高阶导数算法在基本信号中对于照度变化具有敏感性，所以将信号平滑是重要的。强度变化由场景中物体照明的实际变化引起，而不是由图像噪声随机变动所引起。标准的方法，如前面我们在4.1.8.2节所描述的，是用高斯分布函数的卷积：

$$\hat{I}=G\otimes I \tag{4.82}$$

当然，当用一个离散核（如3×3的表）来近似时，其结果本质上是局部的加权平均：

$$G=\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} \tag{4.83}$$

这样的低通滤波器有效地过滤了高频噪声，而且这反过来促使强度的一次导数，特别是使二次导数更为稳定。由于梯度和导数对图像处理的重要性，这种高斯平滑预处理实际上是所有计算机视觉所流行算法的第一步。

4.3.2.1 空间上局部化特征

在计算机视觉团体中，许多算法都假定有意义的物体只占图像的一个子区。所以，被搜索的特征在空间上位于场景图像之内。局部图像处理技术寻找对于像素子集而言是局部的特征，且这种局部特征映射到物理环境的特定位置。这使得它们特别适用于机器人的环境几何模型。

移动机器人学团体使用的最普通的单个局部特征提取器就是边缘检测器。所以，我们以讨论计算机视觉这个经典的题目开始。然而，移动机器人面临着避障和定位这种特别的机动性难题。依照避障，我们提出基于视觉的地平面的提取方法，使机器人能检测到可以安全通过的全部区域。最后，根据定位的需要，我们讨论在机器人导航路标的检测中，基于视觉的特征提取的作用。

边缘检测 图 4.42 表示了一个场景图像，它包含顶灯的一部分以及从该图像提取的边缘。边缘定义了图像平面中发生图像亮度重大变化的区域。如该例所示，边缘检测大大地缩减了图像中的信息总量。所以在图像解释时，是一个有用的潜在特征。假设图像中边缘轮廓对应于重要的场景轮廓，如图 4.42(b)所示，这并不完全真实。边缘检测器的输出和理想直线图存在差别。典型的地方是这里有遗漏的轮廓和噪声轮廓，它们与场景中任何有意义的东西都不相对应。

在图 4.23 中可以看到边缘检测的基本困难。图 4.23(左上)表示一个理想边缘的 1D 部分。但由摄像机产生的信号却更像图 4.23

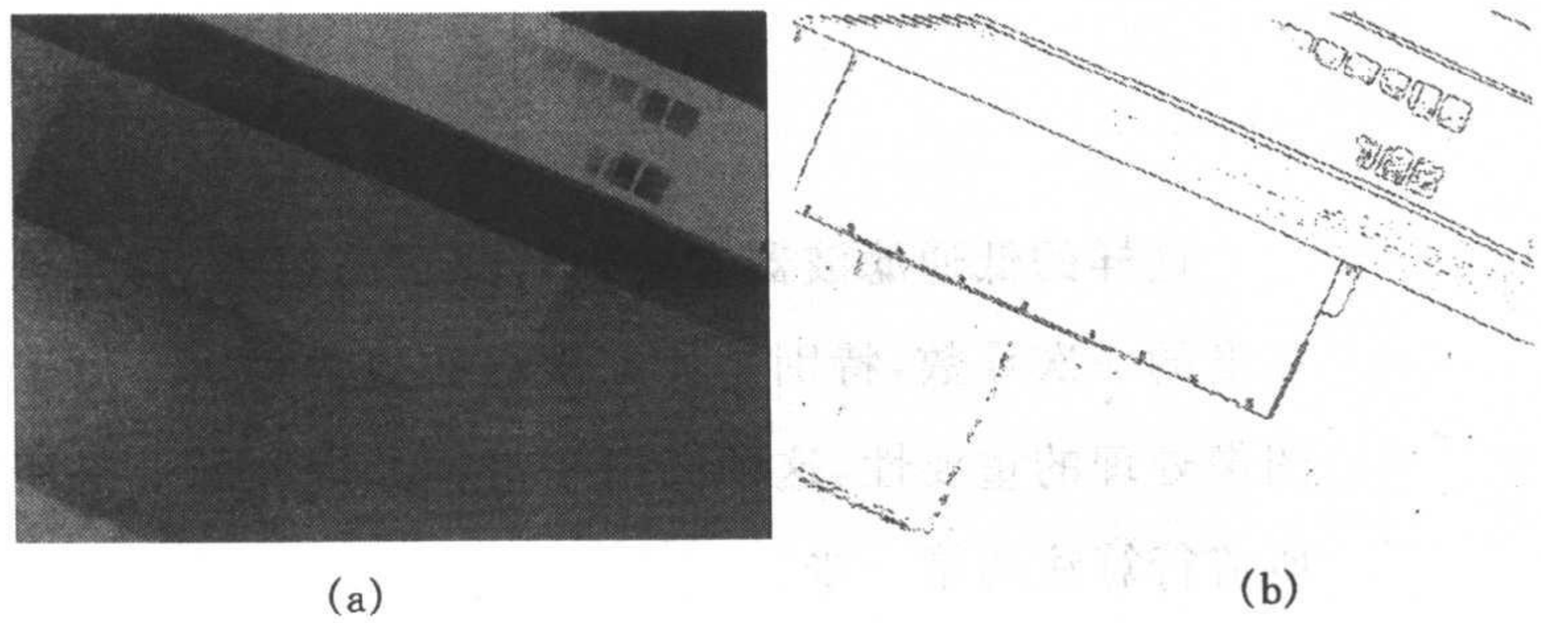


图 4.42 (a) 天花板灯的照片;(b) 从(a)计算的边缘

(右上)。虽然边缘的位置仍在同一个值,但很大电平的高频噪声影响了信号的质量。

新的边缘检测器将完全不一样。因为根据定义,边缘位于强度有大的变化的地方。如图 4.23(右下)所示,有噪声的摄像机信号的微分产生有副作用的峰值,使边缘检测非常困难。使用上面描述的高斯平滑函数将摄像机信号进行预处理,完全可以产生更为稳定的导数。下面提出几个普通的边缘检测算法,所有这些算法都是根据这个基本原理而操作的,即强度的导数遵循某种形式的平滑构成了基本信号,再从中提取边缘特征。

最优边缘检测的 Canny 整个视觉团体,当前参考的边缘检测是由 John Canny 在 1983 年发明的[30]。该边缘检测器出自正规的方法,在这方法里,Canny 把边缘检测当作信号处理问题来对待,它有三个明显的目标:

- 信噪比最大;
- 边缘位置达到最大可能的精确度;
- 与各边缘相关的边缘响应的数目最少。

Canny 边缘检测器将图像经高斯卷积平滑后在(被校正的)导数中求最大值。在实践中,平滑和微分被合成一个操作,因为

$$(G \otimes I)' = G' \otimes I \tag{4.84}$$

因此，通过与高斯 G_σ 进行卷积将图像平滑，然后求导，等效于用 G'_σ 将图像进行卷积，G'_σ 是高斯的一阶导数(图 4.43(b))。

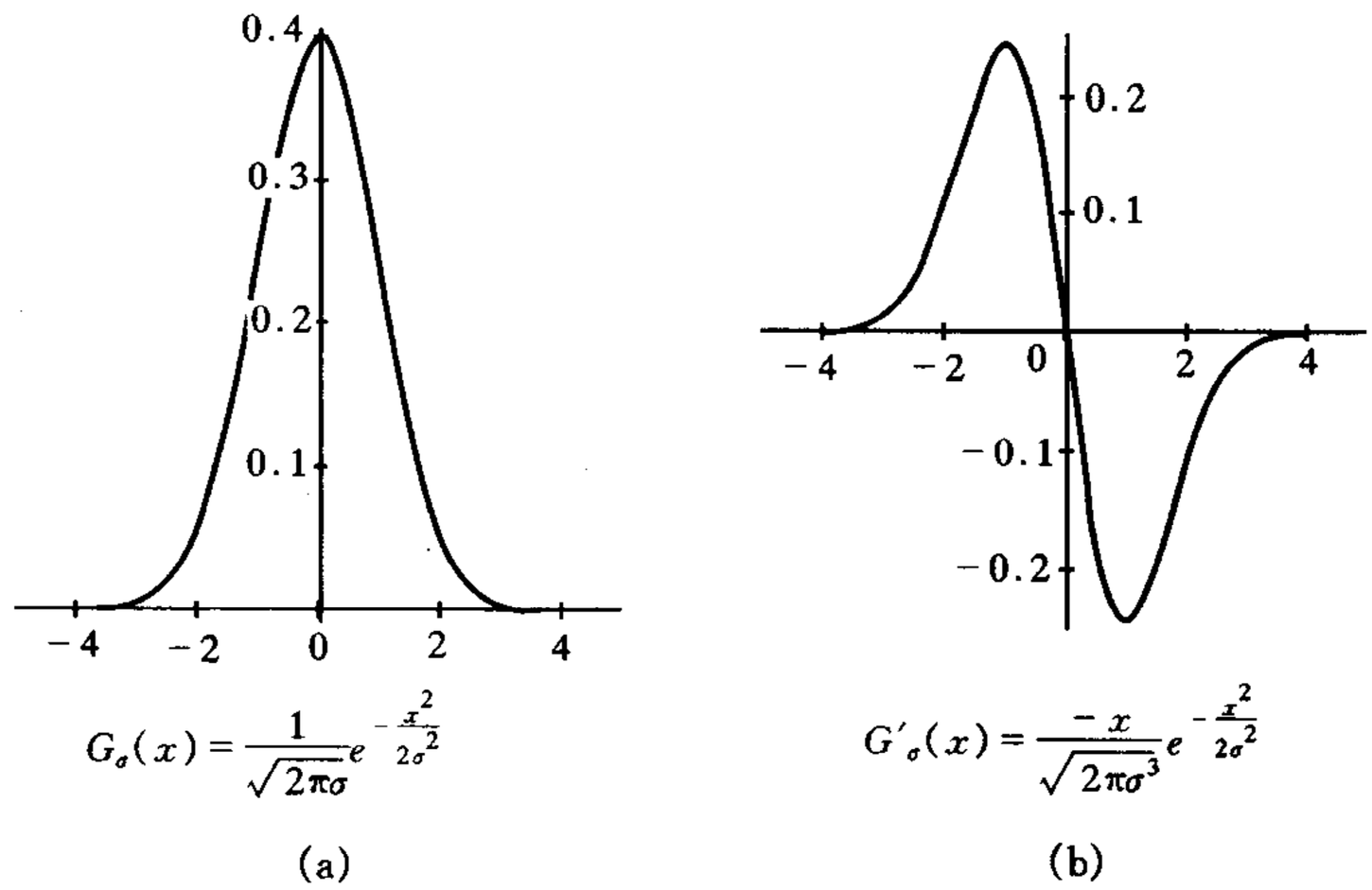


图 4.43 (a) 高斯函数；(b) 高斯函数的一阶导数

我们希望在任何方向上检测边缘。因为 G' 是有方向的，正像在方程(4.35)中我们进行拉氏运算一样，这要求应用两个正交的滤波器。我们定义这两个滤波器为 $f_V(x,y)=G'_\sigma(x)G_\sigma(y)$ 和 $f_H(x,y)=G'_\sigma(y)G_\sigma(x)$。结果是一个在任意方向检测边缘的基本算法。

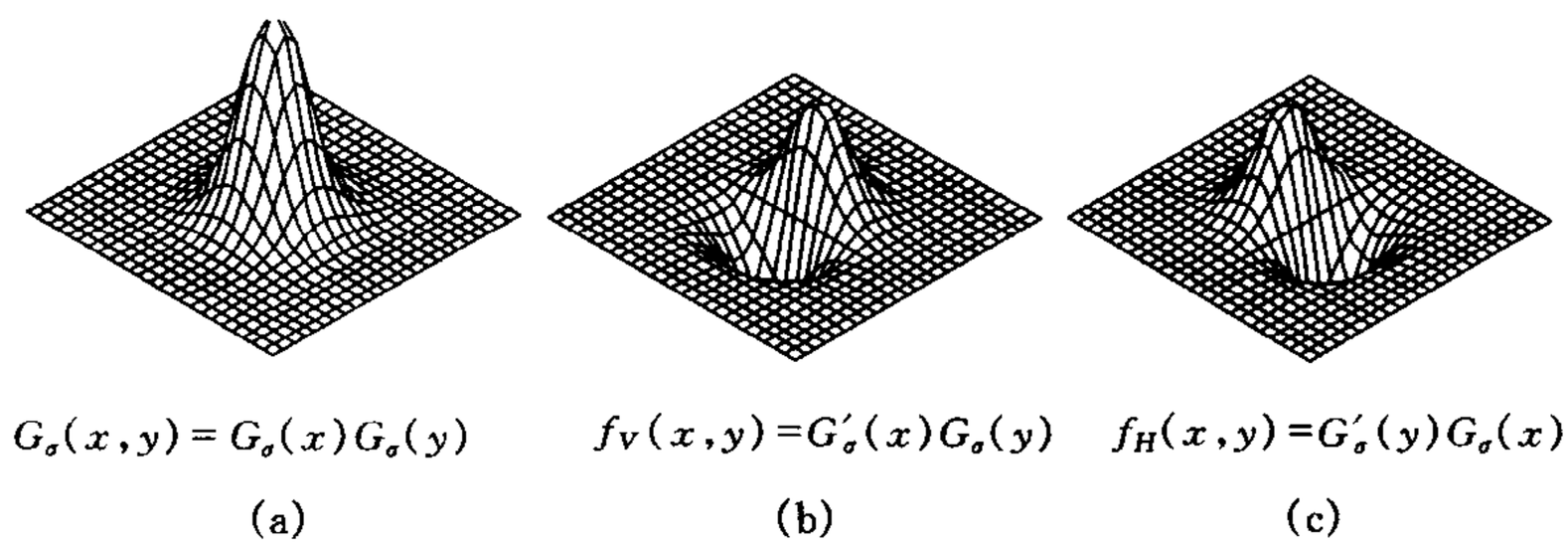


图 4.44 (a) 二维高斯函数；(b) 垂直滤波器；(c) 水平滤波器

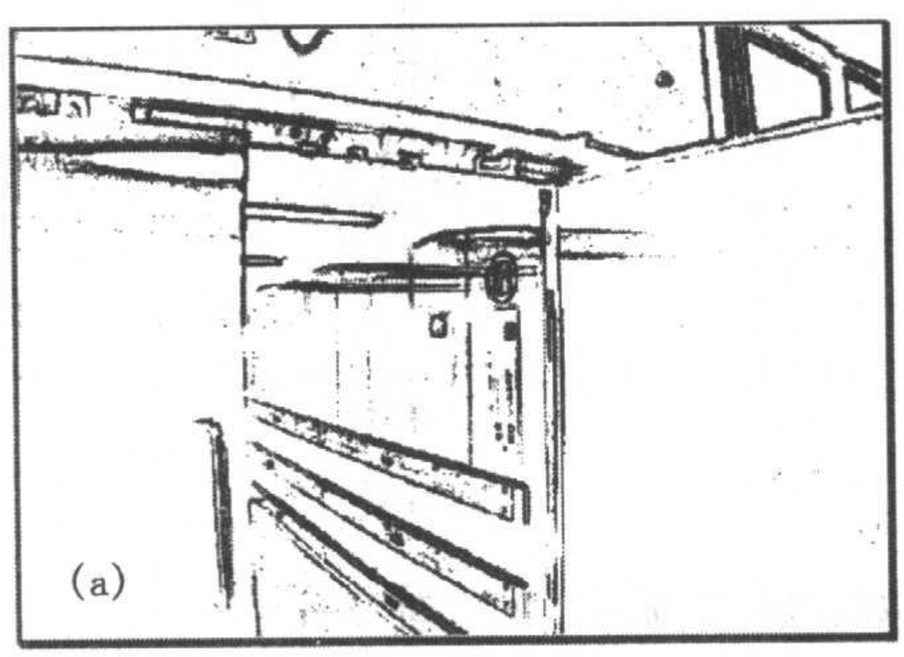
在任意方向检测边缘像素的算法如下：

(1) 对图像 $I(x,y)$ 和 $f_V(x,y)$ 与 $f_H(x,y)$ 进行卷积，分别得到梯度分量 $R_V(x,y)$ 和 $R_H(x,y)$；

(2) 定义梯度幅值的平方 $R(x,y)=R_V^2(x,y)+R_H^2(x,y)$；

(3) 将那些在 $R(x,y)$ 中高于预定阈值的峰值作标记。

一旦提取了像素，下一步是构建完整的边缘。在这个过程中，通用的下一个步骤是**非最大压缩**。其过程包括利用边缘的方向信息修正梯度值并确定它是否是局部最大。如果不是，将值置为零。这使得只保留最大值，因而减小了单个像素所有边缘的厚度(图 4.45)。



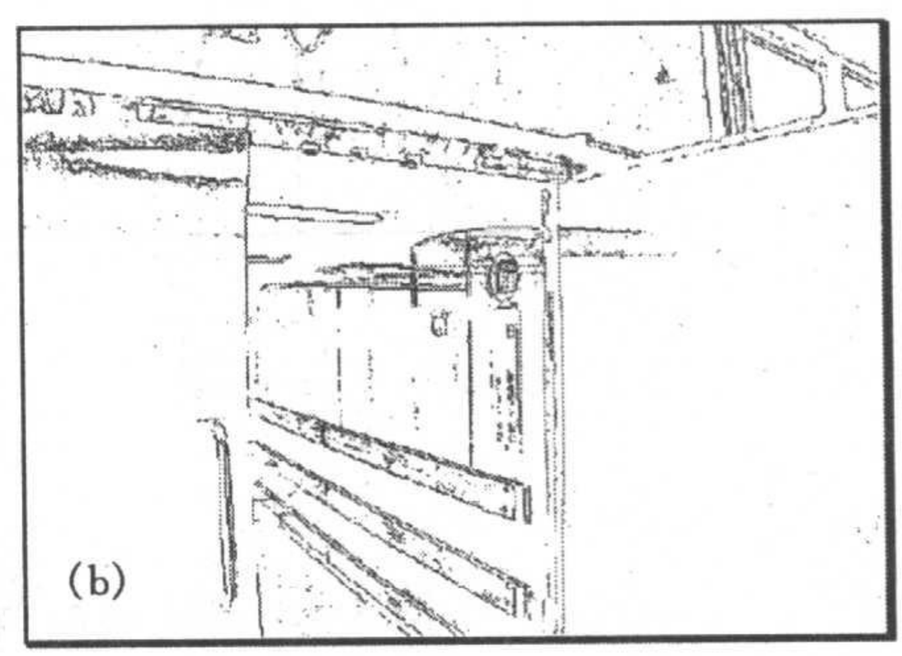


图 4.45 (a) 边缘图像的例子；(b)是(a)的非最大压缩

最后，我们已经准备好从边缘像素获得整个边缘。第一，寻找相邻(或相连)的边缘集合，并将它们聚合成有序的列表。第二，用阈值消去最弱的边缘。

梯度边缘检测器 在移动机器人上，必须使计算时间最少，以保持机器人的实时行为。所以，常用比较简单的离散算子逼近 Canny 边缘检测器的行为。Roberts 在 1965 年[29]开发了这种早期的算子。他使用了两个 2×2 掩码在两个对角线方向计算跨边缘的梯度。令 r_1 为从第一个掩码算出来的值，r_2 是从第二个掩码算出来的值。Robert 用以下方程得到梯度幅值 $|G|$：

$$|G|=\sqrt{r_1^2+r_2^2};\quad r_1=\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix};\quad r_2=\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix} \tag{4.85}$$

Prewitt(1970)[29]使用两个面向行和列方向的 3×3 掩码。令 p_1 为从第一个掩码计算而得的值，p_2 为从第二个掩码计算而得的

值。如下面方程所示，Prewitt 得到梯度幅值 $|G|$ 和相对于列轴按顺时针的角度所取的梯度方向 θ。

$$|G| \approx \sqrt{p_1^2 + p_2^2};$$

$$\theta \approx \arctan\left(\frac{p_1}{p_2}\right); \quad p_1 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}; \quad p_2 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{4.86}$$

在同一年，和 Prewitt 一样，Sobel[29] 使用两个面向行和列方向的 3×3 掩码。令 s_1 为从第一个掩码计算而得的值，s_2 为从第二个掩码计算而得的值。Sobel 得到与 Prewitt 相同结果的梯度幅值 $|G|$ 和相对于列轴按顺时针的角度所取的梯度方向 θ。图 4.46 表示了

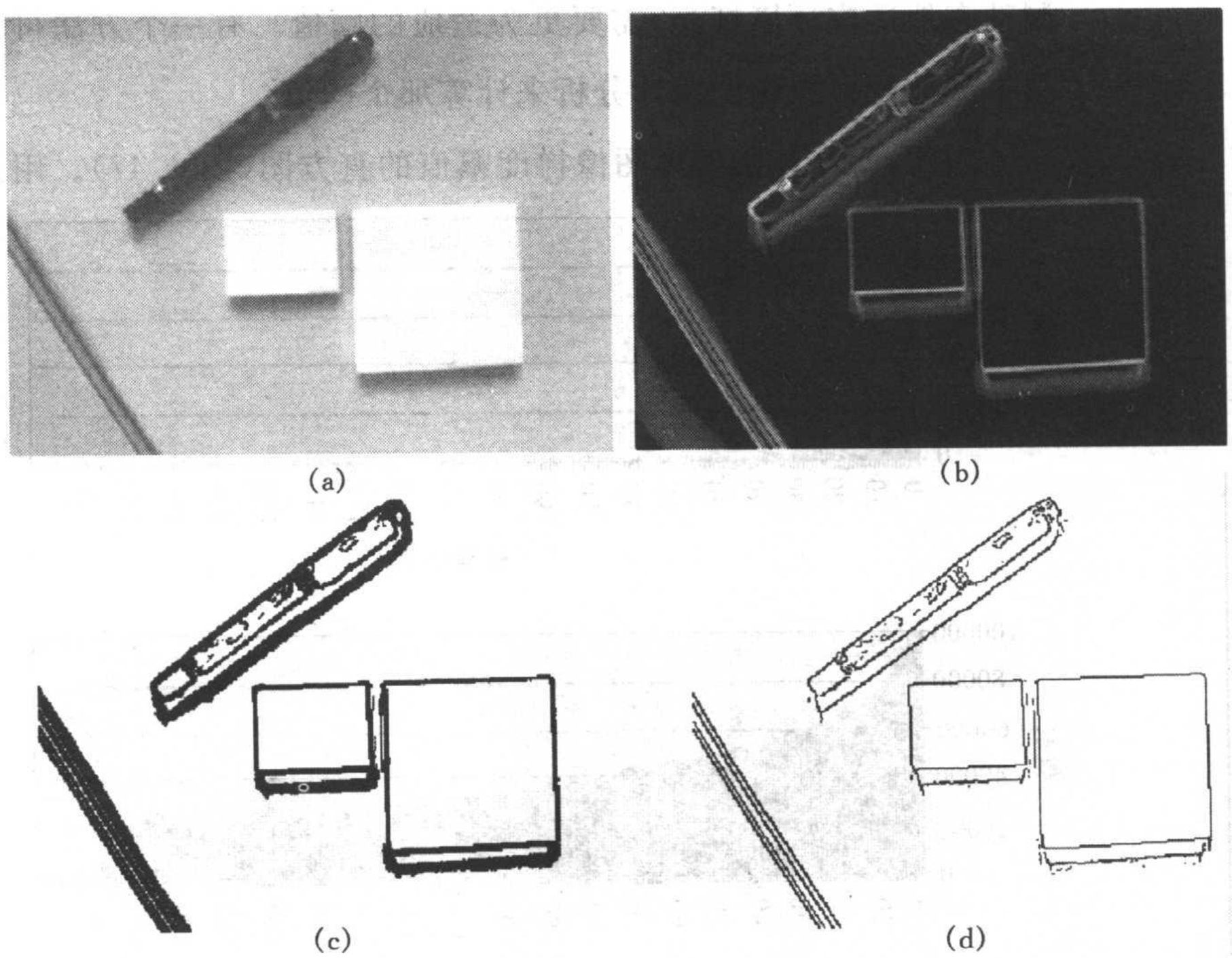


图 4.46 用不同的处理步骤，基于视觉特征提取的实例：(a) 原始图像数据；(b) 用 Sobel 滤波器的滤波图像；(c) 加阈值，边缘像素的选择；(d) 非最大压缩

Sobel 滤波器应用于可视场景。

$$|G| \approx \sqrt{s_1^2 + s_2^2};$$

$$\theta \approx \arctan\left(\frac{s_1}{s_2}\right); \quad s_1 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}; \quad s_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{4.87}$$

动态设阈值 许多图像处理算法一般已在实验室条件或用静态图像数据库做过测试。然而，移动机器人运行在动态实时的背景下，关于最优或甚至稳定的照明并无保证。移动机器人的视觉系统必须适应于变化的照明，所以边缘检测的恒定阈值电平是不合适的。具有不同照明的同一场景会产生巨大差异的边缘图像，为了使边缘检测动态地适应环境灯光，需要更为适应的阈值。有一个方法可以专门根据被处理图像的统计分析来计算那个阈值。

为此，要计算被处理图像梯度幅值的直方图(图 4.47)。用这个

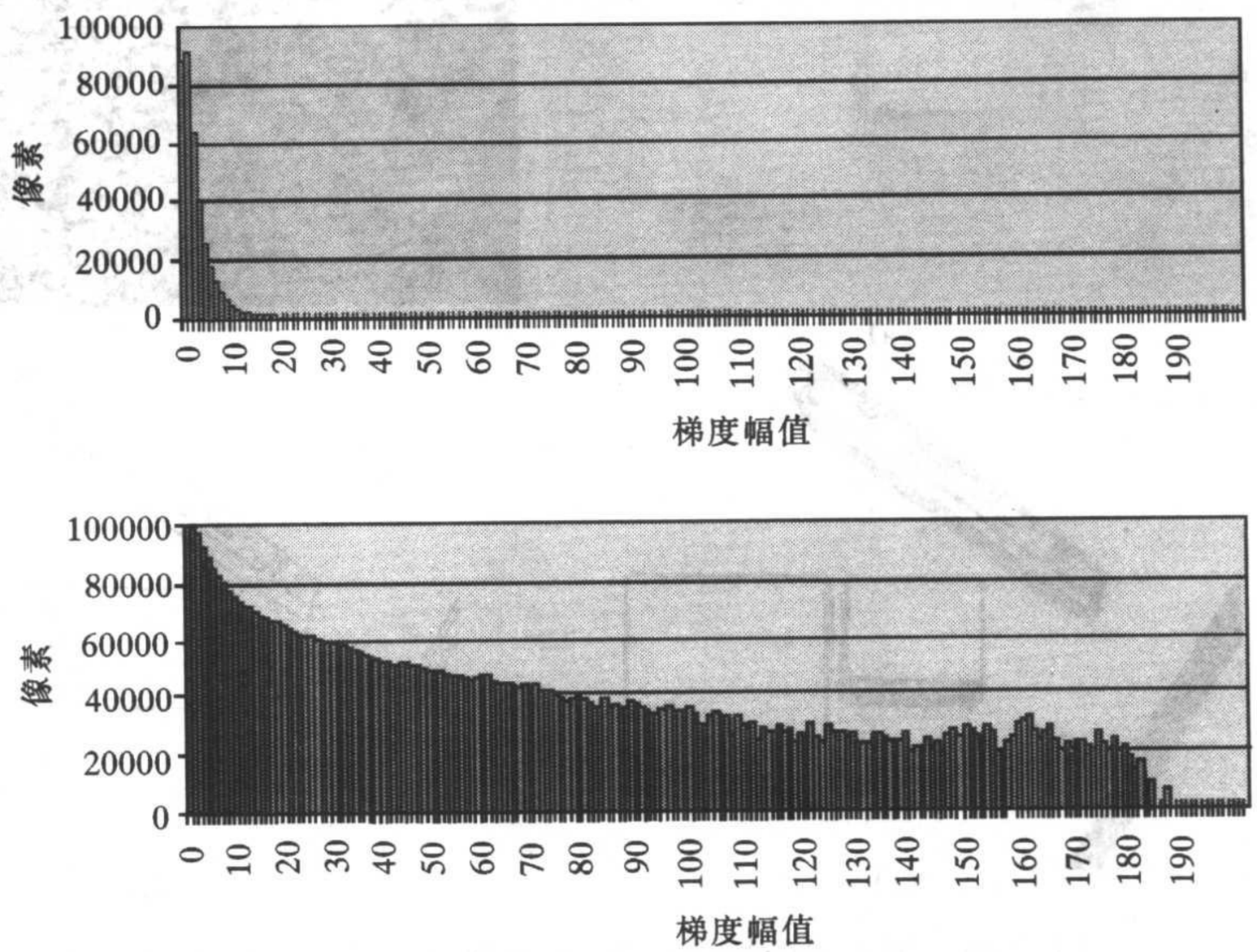


图 4.47 (a) 图 4.46(b)中具有特定值的像素数目；(b)与(a)一样，但用对数标度

简单的直方图，容易只考虑 n 个具有最高幅值的像素来做进一步的计算。从最高梯度幅值开始，向后计数像素，达到 n 的那一点，其梯度幅值被用作临时的阈值。

这个计算的动机是：具有最高梯度的 n 个像素希望成为被处理图像最相关的部分。而且，对各图像，要考虑相同数目的有关边缘的像素，它与照明无关。重要的是要对以下事实引起注意：由边缘检测器提交的边缘图像像素数目不是 n。因为大多数检测器使用非最大压缩，边缘像素的数目将会被进一步减少。

直线边缘提取：霍夫变换 在移动机器人学中，直线边缘常常被提取为特定的直线。例如，直线垂直边缘可以用作门口的位置和走廊交叉口的提示。霍夫变换是提取特殊形式边缘的简单工具[16,18]，这里我们阐述它在提取直线边缘中的应用。

假定在图像 I 中的像素 (x_p, y_p) 是边缘的一部分，任何包括点 (x_p, y_p) 的直线边缘必须满足方程 $y_p = m_1 x_p + b_1$。该方程只能用受约束的 m_1 和 b_1 的可能值集合予以满足。换句话说，该方程只能用通过 (x_p, y_p) 经 I 的直线予以满足。

现在我们来看看在 I 中的第二像素 (x_q, y_q)。任何经过这第两点的直线必须满足方程 $y_q = m_2 x_q + b_2$，如果 $m_1 = m_2$ 和 $b_1 = b_2$ 会怎么样呢？则由两个方程定义的直线就是一条相同的直线：这是通过 (x_p, y_p) 和 (x_q, y_q) 两点的直线。

一般来说，对经 I 的单条直线部分的所有像素，它们必全部处于由**相同的** m 和 b 值所定义的一条直线上。当然，该直线的通用定义是 $y = mx + b$。霍夫变换利用这个基本性质建立了一个机制，使各边缘像素能够对参数 (m, b) 不同的值“投票”。最后，具有最多选票的直线就是直线边缘特征：

- 建立 2D 阵列 A，A 具有 m 和 b 值棋盘格化的轴；
- 阵列初始化为零：对所有 m 和 b 值，$A[m, b] = 0$；
- 对 I 中各边缘像素 (x_p, y_p)，对所有 m 和 b 值，循环：

如果 $y_p = mx_p + b$，则 $A[m,b] += 1$；

- 在 A 中搜索单元，辨识那些具有最大值的单元。各个这种单元的指数(m,b)对应于 I 中一个被提取的直线边缘。

地平面的提取 避障是移动机器人所要求的基本任务之一。基于距离的传感器为辨识移动机器人面临的障碍提供了有效的方法。事实上，因为它们直接测量环境中距物体的距离，像超声、激光测距仪这样基于距离的传感器本质上很适合于障碍的检测。但各测距传感器有局限性。超声角度分辨率差，且在浅表角遭受相干反射。大多数激光测距仪是 2D 的，只检测穿透特定被感知平面的障碍物。为了能分别给出对应性和模糊度，立体视觉和景深需要障碍物和地平面具有纹理。

除了各个单独的缺点之外，基于距离的障碍检测系统，检测在地上小的或平坦的障碍物有困难。例如，真空吸尘器(机器人)也许需要避开大或平的物体，如弃在地上的纸或钱币。另外，不同类型的地面表面不容易被测距区分。例如，人行道行走的机器人仅仅用距离感知，则从人行道区分出草地会有困难。

地平面提取是辨识地面可行走部分的一种基于视觉的方法。因为在不同种类的实现方法中，它使用了边缘和颜色，所以这种障碍检测系统在传统测距装置有困难的场合可以容易地检测障碍。

在所有用基于视觉算法的情况下，地平面提取只在满足以下几个重要假定的环境下才能成功：

- 障碍在表象上与地面有差别；
- 地面是平的，它与摄像机的角度已知；
- 没有悬挂的障碍物。

为了用表象区分地面和障碍物，第一个假设是一个需求。有时要用该假设的更强形式规定地面在表象上是均匀的，并与所有的障碍物不同，第二和第三个假设允许地平面提取算法能估计机器人距所测障碍物的距离。

在人工环境中地面平面的提取 在一个受控的环境中，地面、墙和障碍物可以被设计得使墙和障碍物与摄像机图像中地面表现出有重大的差别。从 1966 年到 1972 年在 SRI(斯坦福研究院)开发了第一个自主机器人 Shakey，它在一个生产制造环境中用基于视觉的地面提取技术作障碍检测[115]。Shakey 的人工环境采用无纹理的、均匀的白色地面瓷砖。而且，各墙的基部用高对比度的黑漆条进行油漆，所有简单的多边形障碍物的边缘也漆成黑色。

在 Shakey 的环境中，对应于非地面物体的边缘以及地平面的提取算法，是由边缘检测器应用于单色摄像机的图像简单地组成。图像中检测到的最低边缘对应于最近的障碍物，从图像中提取的直线边缘不仅提供了关于位置的提示，也提供了墙和多边形障碍物的方向。

虽然这个简单的基于表象的障碍检测系统是成功的，但应该注意到，在实验室建立间接照明的时刻必须采取特别的处理使之不投射阴影，因为系统会错误地把阴影的边缘解释成障碍物。

自适应的地平面的提取 地平面提取不仅在人工环境中获得成功，而且在现实环境的移动机器人示范中也获得成功。在示范中，机器人既避开了静态障碍，如墙；又以几个赫兹(Hz)的速率，根据地面平面的分割避开了动态障碍，如过路的行人。这种地平面提取算法倾向于联合使用边缘检测和颜色检测，与此同时作了有关地面模型的假定，如地面的最大纹理或近似的颜色范围[78]。

基于有关地面表象的固定假设的各种系统，只限定于那些满足其约束的环境。更新近的方法是自适应的地平面的提取。籍此，它允许定义地面期望表象的参数随时改变。在最简单的情况下，人们可以假定图像底部的像素(即最靠近机器人)是地面部分，且不包含障碍物。然后，对这些“地面样本”所计算的统计特性可以被用来分类其余的图像像素。

自适应系统关键性的挑战是利用“地面样本”像素，选择什么样

的统计特性来计算。最普通的解决方法是构建一个或多个基于地面样本像素值的**直方图**。在上述“边缘检测”条件下，我们发现在边缘检测设阈值的算法中，在决定最佳分割点时，直方图是有用的。直方图作为分布的离散表示也是有用的。与高斯表示不一样，直方图可以获得多模态分布，也可以快速更新直方图且使用很少的处理器内存。构造一个图像 I“地面样本”的子区 I_f 的强度直方图的方法如下：

- 作为预处理，用高斯平滑算子，平滑 I_f；
- 用 n 个强度值，将直方图阵列 H 初始化，对 $i=1,\cdots,n$；$H[i]=0$；
- 对每一个在 I_f 的像素 (x,y) 递增直方图：$H[I_f(x,y)]+=1$。

直方图阵列 H 用作地平面表象的特征化。通常对应于强度、色调和饱和度构建几个 1D 的直方图。在 I 中各像素分类成地平面或障碍，是通过对目标像素的质量查看合适的直方图计数予以实施的。例如，如果目标像素有一个在“地面样本”从不发生的色调，那么相应的色调直方图将有零的计数。当像素参考的直方图值低于预定的阈值时，像素被分类为一个障碍。

图 4.48 表示了操作在室内和室外图像[151]上的一个基于表象的

图 4.48 自适应地面平面提取的例子。梯形多边形辨识地面采样区

地平面提取算法。注意，与静态地面提取算法不同，由于自适应的直方图表示，自适应算法能够成功地区分人类阴影。研究工作的一个有意义的扩展是不用静态地面采样的假定，而是记录可视的历史数据作为地面采样，而且只采用先验的视觉图像部分，该部分是在移动机器人运动期间，机器人已成功地拍录的。

基于表象的地平面提取已在室内和室外两种环境作实时避障示范，带宽高至 10 Hz。应用包括机器人割草、室内社交机器人和自动电气轮椅。

4.3.2.2 全图像特征

一个单独可视图像能够提供如此多的关于机器人即时环境的信息，于是搜索图像特殊局部特征的另一个办法可以利用整个图像所获取的信息来提取全图像特征。全图像特征不是被设计成识别指定的空间结构，诸如障碍物或特定路标的位置，而是把它们用作整个局部区域的精确表示。从机器人定位的观点看，其目标是从与机器人位置密切相关的图像中提取一个或多个特征。换句话说，机器人位置小的改变应只造成全图像特征小的变化，同时机器人位置大的改变应相应地使全图像特征有大的变化。

下面我们提出全图像特征提取的两种技术。第一个技术是图像直方图方法的另一个普遍的应用，最终的图像直方图组成了一组全图像特征，特征直接从图像像素信息中推导出来；第二种技术，即层次提取，它覆盖了许多方法。在这些方法里，首先通过提取空间上局部的特征建立全图像特征，然后把这些特征组合在一起形成一个单独的元特征。

直接提取：图像直方图 记得我们希望设计全图像特征，它对机器人小量的运动不敏感，同时对大规模的机器人移动记录重大的变动。为达到这个目的，在设计基于视觉的传感器时，逻辑上的第一步是使摄像机的视场最大化。当视场增大时，机器人环境中小尺寸的结构占据图像的较小部分，减轻了单独场景物对图像特征的影响。

现在，在移动机器人学中非常普遍的反折射的摄像机系统具有极广的视角[114]。这个成像系统由一个朝向抛物镜安装的高质量CCD摄像机和专用的光学系统组成。如图4.49所示，图像提供机器人环境的360°视角。

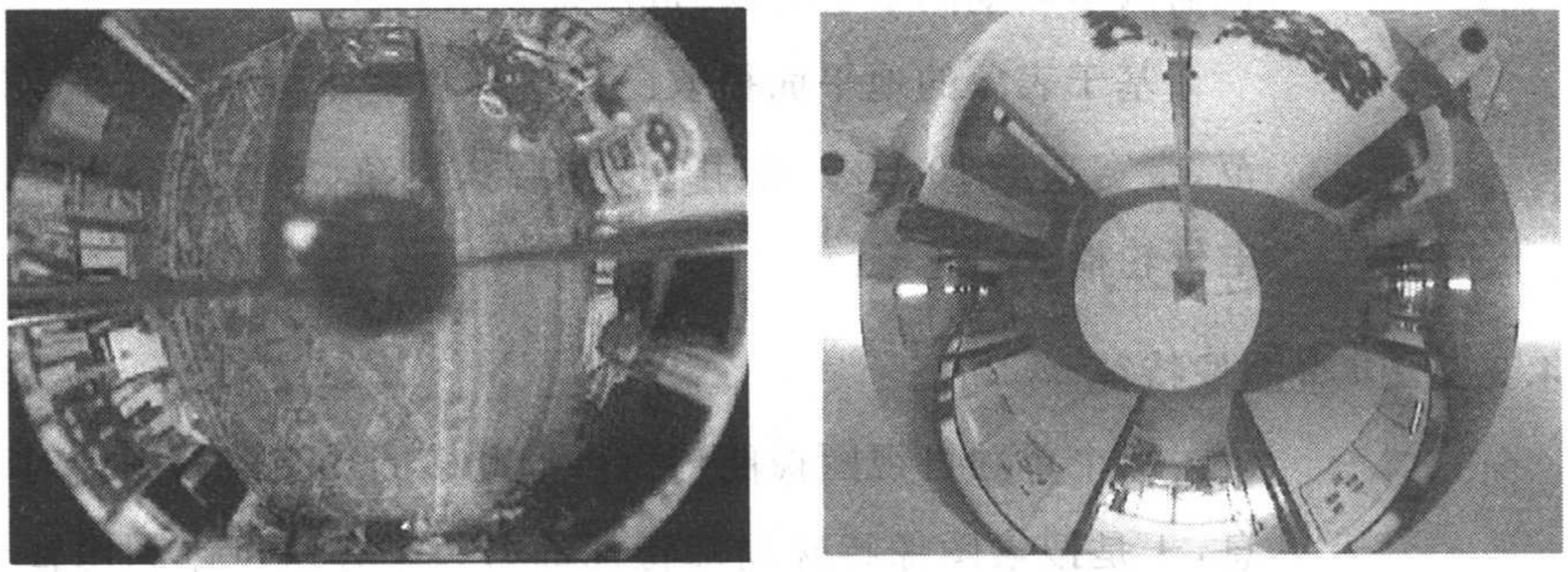


图4.49 由OmniCam反折射的摄像机系统获取的两个典型图像

反折射的图像是扭曲在2D表面上的360°图像。因此，根据对小尺寸机器人运动的灵敏度，它具有另一个重要优点。如果把摄像机垂直地装在机器人上，使图像表示围绕机器人的环境（即它的范围），摄像机和机器人的转动只产生图像的转动。简而言之，反折射的摄像机在旋转上可以是对视场不变。

当然，移动机器人仍然改变了图像，即像素位置将会改变，虽然新的图像只是原图像的转动。但我们打算经直方图提取图像特征，因为直方图编制是像素值集合的函数，不是各像素位置的函数，所以该过程是像素位置不变。当与反折射的摄像机的视场不变性结合在一起时，我们可以建立一个对机器人转动不变的系统，且它对小尺寸机器人平移不敏感。

彩色摄像机的输出图像依照多个**波段**一般包含有用的信息：r、g和b值以及色调、饱和度和亮度值。最简单的基于直方图的提取策略是建立表征各波段的各自的1D直方图。给定一个彩色摄像机图像G，第一步是建立从G到各n个可用波段的映射。我们用G_i当作

G 所有像素中，存储波段 i 值的阵列。各指定波段的直方图 H_i，计算如前：

- 作为预处理，用高斯平滑算子，平滑 G_i；
- 用 n 个电平，使 H 初始化，对 $j=1,\cdots,n$，$H[j]=0$；
- 对每一个在 G_i 的像素 (x,y) 递增直方图：$H_i[G_i(x,y)]+=1$。

给定图 4.49 的图像，图像直方图技术提取 6 个直方图（对各 r、g、b、色调、饱和度和亮度），如图 4.50 所示。为了利用这种直方图作

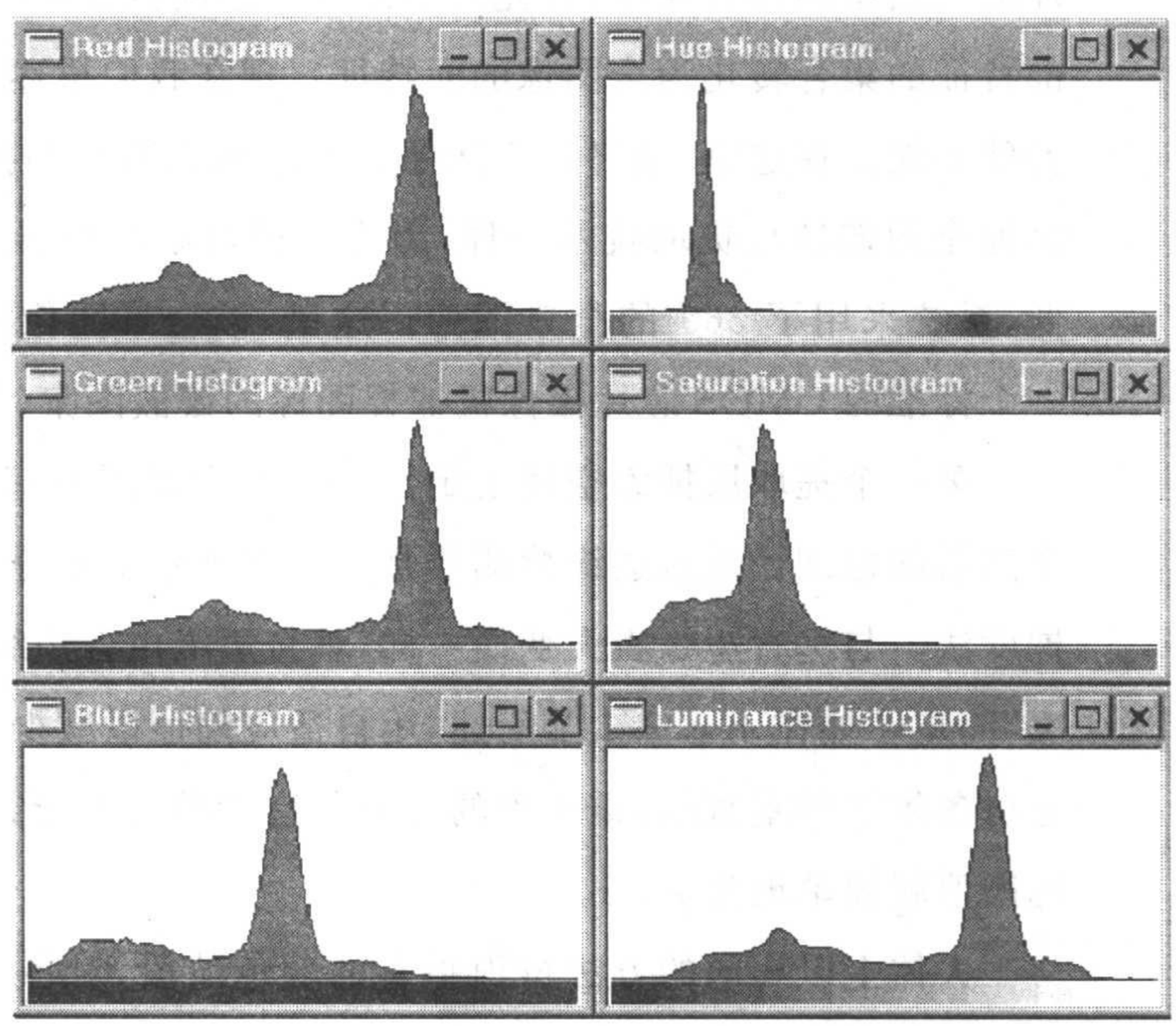


图 4.50 上面是图像的 6 个 1D 直方图，在编制直方图之前，5×5 平滑滤波器与各波段进行卷积

为全图像特征，我们需要多种途径比较直方图，对直方图映射到附近机器人位置的似然值进行量化。定义有用的直方图距离度量，其本身是图像检索领域内一个重要的子领域。作为综述，请参阅文献[127]。在移动机器人定位中所碰到的最成功距离度量之一是**杰弗里散度**(Jeffrey divergence)。给定具有表示直方图实体 h_i 和 k_i 的两个直方图 H 和 K，杰弗里散度 $d(H,K)$ 定义为：

$$d(H,K)=\sum_i\left(h_i\log\frac{2h_i}{h_i+k_i}+k_i\log\frac{2k_i}{h_i+k_i}\right) \tag{4.88}$$

利用诸如杰弗里散度这样的尺度，移动机器人已经用全图像直方图特征对以前记录的环境位置的图像数据库实时辨识它们的位置。利用这个全图像提取方法，机器人可以容易地恢复它所处的特殊走廊或特殊房间[152]。

分层提取：图像指印提取 另一个直接从像素值提取全图像特征的方法是利用分层法：首先辨识图像中空间位置特征，然后将该局部特征的集合转化为全图像的元特征。现在我们描述该方法的一个特殊实现。在这里，最终的全图像特征被称为图像的**指印**[95]。像用其他全图像特征提取技术一样，因为希望对小的机器人运动灵敏度低，系统采用了360°的全景图像。这里，该全景图像构造成一个用一个标准的CMOS芯片摄像机捕捉图像的镶嵌图案(马赛克)。

第一个提取层搜索空间上定位特征的全景图像：垂直边缘和16个离散的色调。垂直边缘检测器是一个实现水平差分算子的简单的梯度法。与垂直边缘霍夫变换一样，垂直边缘由各边缘像素“投票表决”而定。如在4.3.2.1节所述，用自适应阈值以减少边缘数目。假定对各候选的垂直线，霍夫表的计分具有均值μ和标准偏差σ，则所选阈值就简单地为$\mu+\sigma$。

大体上以相同的方法辨识垂直的颜色波段，对各颜色的出现确定统计，然后，除了那些计分大于$\mu+\sigma$之外，筛选出所有侯选颜色的斑纹。图4.51表示了两个样本全景图像和它们相关的指印。注意，各指印被转换成用ASCII字符串表示。

正像在图像直方图编制中使用直方图的距离度量一样，我们需要对两个指印之间的距离有一个可定量的度量。但字符串匹配算法是另一个大的研究领域，今天在基因[34]领域有特别重要的应用。注意，我们拥有的字符串，不仅单个元素值有差别，甚至它们整个长度也有差别。例如，图4.52表示了用上面算法所产生的三个实际序列。头一个字符串应该与Place 1相匹配。但我们注意到，在字符串

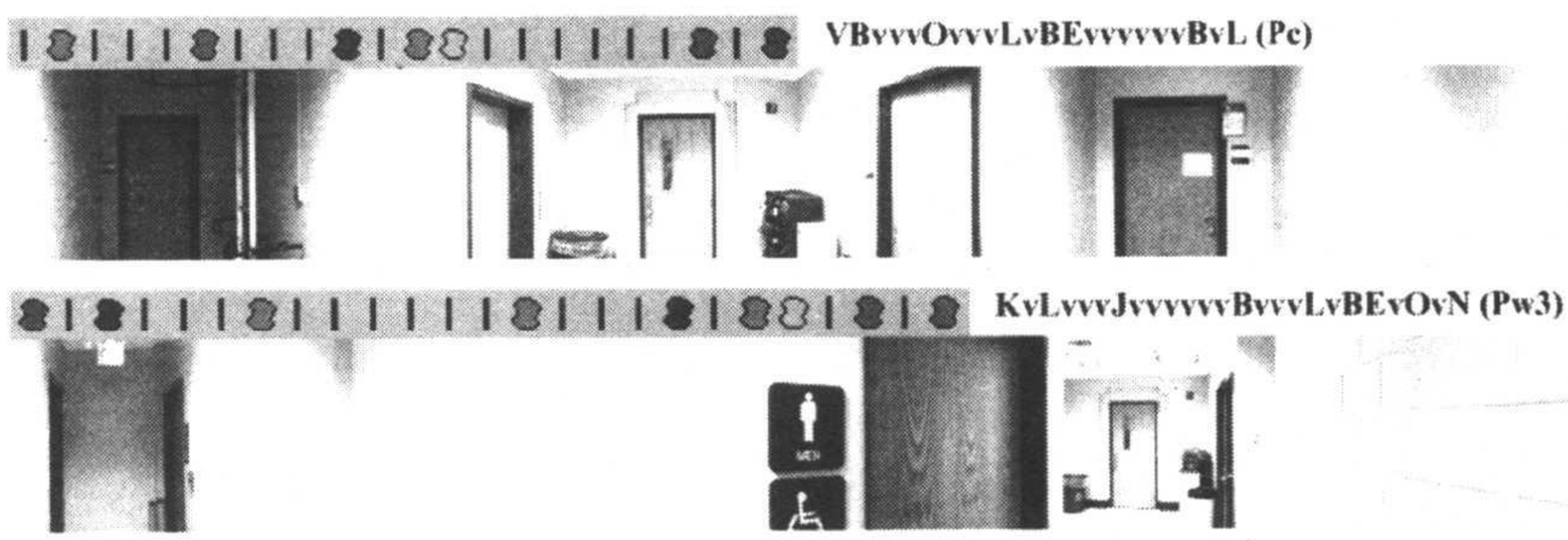


图 4.51 两个全景图和它们相关的指印序列

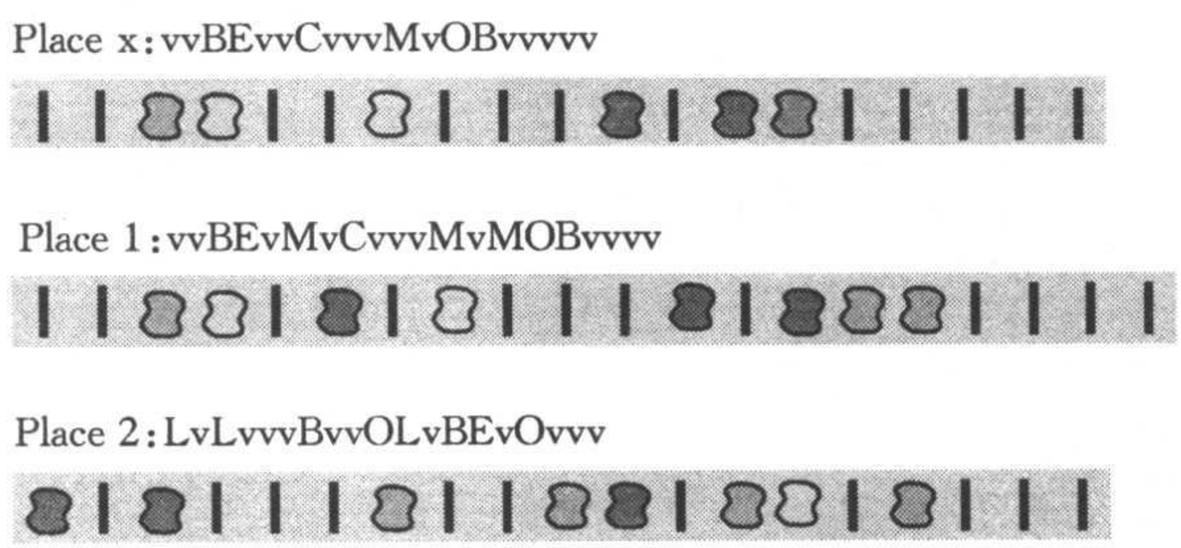


图 4.52 三个实际符号串序列。头两个是在同一位置上由机器人提取的符号串

之间存在删节和插入。

用于字符串区分的指印法技术被称为**最小能量算法**。借鉴立体视觉界的思想，该基于优化的算法会找到一个序列“转换”成另一个序列所需的最小能量。其结果是一个距离的度量，它对增加或减少单个局部特征相对地不敏感度，同时在不同的环境中仍能鲁棒地确认正确匹配的字符串。

从前两节中，应该清楚，全图像特征提取用基于视觉的感知是直接明了的，且可以应用于移动机器人的定位。由于它们直接应用于更迫切需要的实时避障，所以这是一个继续起支配作用的空间上的局部特征。

第 5 章　移动机器人的定位

5.1　引言

导航是对移动机器人所要求的最具挑战性的能力之一。导航的成功需要导航的四个模块的成功：**感知**，机器人必须解释它的传感器信息，提取有意义的数据；**定位**，机器人必须确定它在环境中的位置（图 5.1）；**认知**，机器人必须决定如何行动以达到目标；**运动控制**，机器人必须调节它的运动输出，以实现期望的轨迹。

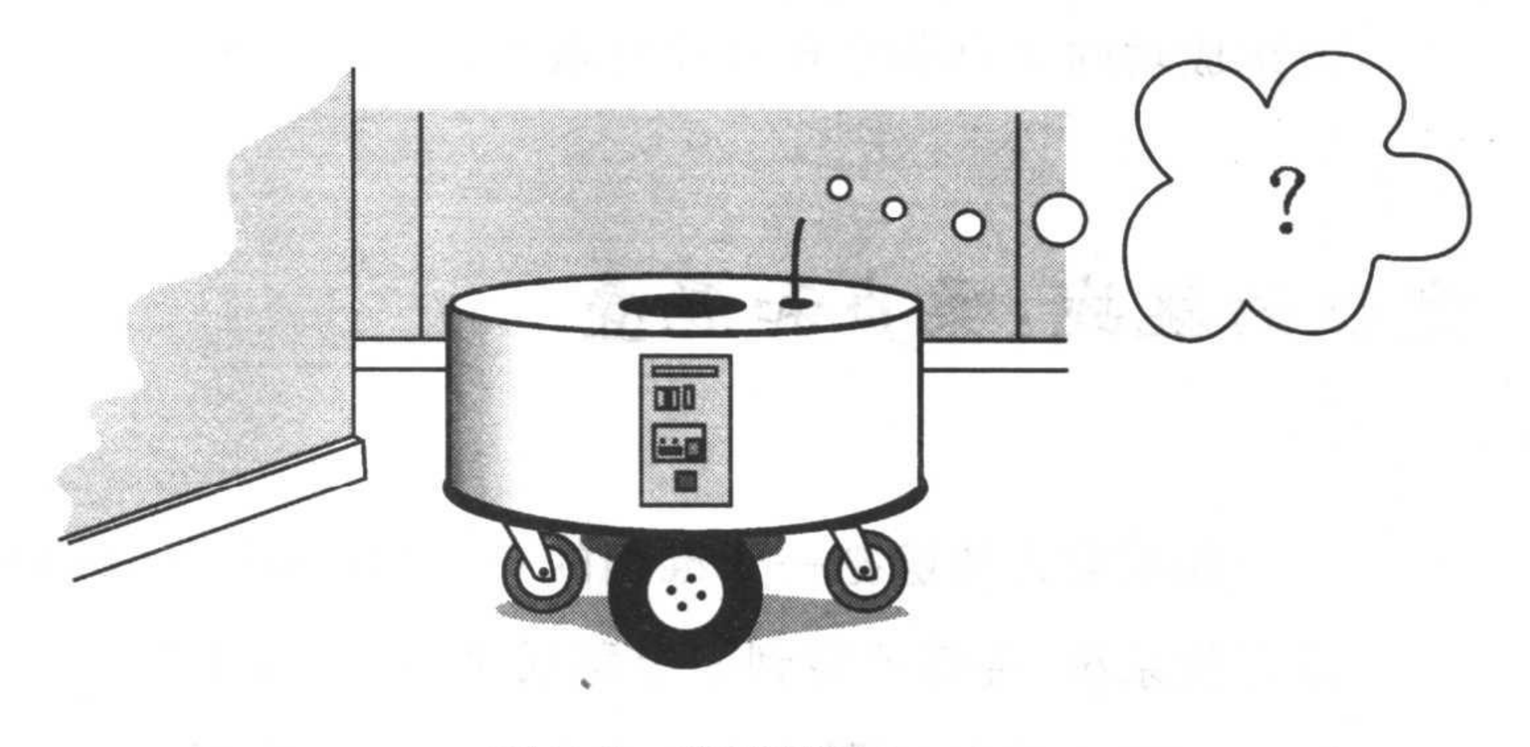


图 5.1　我在哪里？

在这四个模块中（图 5.2），定位研究在以往的几十年里受到最大的关注。因此，这方面已取得重大的进展。本章我们阐述最近几年成功的定位方法学。首先，5.2 节描述传感器和执行器的不确定性如何造成定位的困难。然后，在 5.3 节，描述两个极端的方法以处

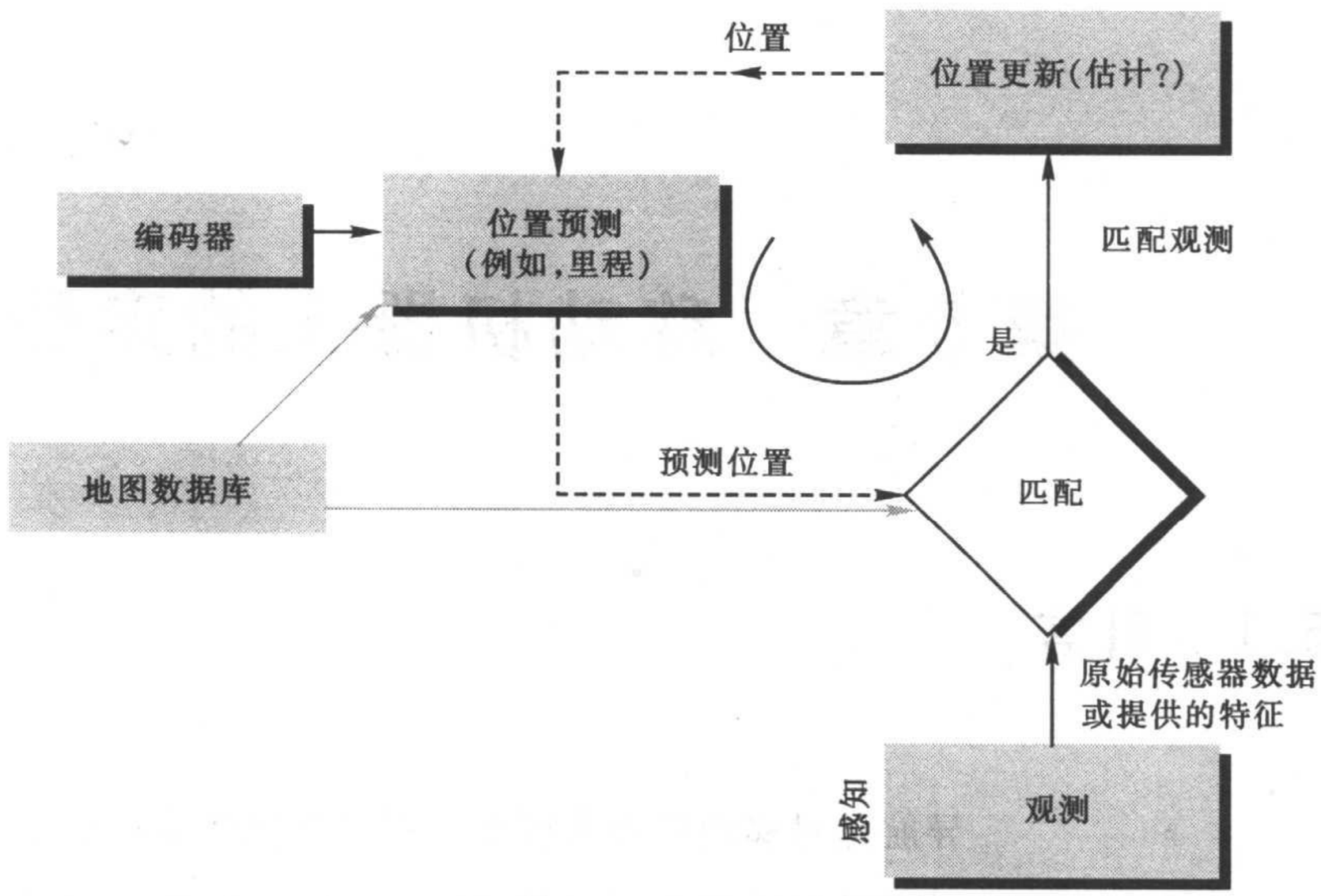


图 5.2 移动机器人的一般框图

理机器人定位的困难:完全取消定位和执行明显的基于地图的定位。本章的其余部分讨论表示的问题,并利用不同的表示方法,提出定位系统成功的实例和实现移动机器人定位的技术。

5.2 定位的挑战:噪声和混叠

如果有人能够把一个准确的 GPS(global positioning system,全球定位系统)传感器装到一个移动机器人上,许多定位问题就可以排除。GPS 会通知机器人它在室内和室外的准确位置,所以常常会立即得到"我在哪里?"的回答。遗憾的是,这样的传感器当前是不切实际的。现有的 GPS 网络提供的准确度在几米之内,这种准确度对定位人一样大小的移动机器人和微型机器人(如台式机器人)以及未来体内驾驶的毫微机器人是不能接受的。而且 GPS 技术不能在室内或者有障碍的区域起作用,因此限制了它的工作空间。

但是,除了 GPS 限制之外,我们可以看出,定位不仅仅意味着要知道在地球参考系中物体的绝对位置。考虑到与人交互的一个机器人,也许需要确认它的绝对位置。但是它相对于目标人物的相对位置也同样重要。它的定位任务包括用它的传感器阵列辨识人类,然后计算它与人的相对位置。而且,在认知期间,为了达到目标,机器人会选择一种策略。如果它企图到达一个特定的位置,那定位也许还不够。机器人也许要获取或建立一个环境模型,即一张地图,它帮助机器人规划一条达到目标的路径。再者,定位意味着不只是简单地确定空间中一个绝对姿态,它意味构建一张地图,而后确定机器人相对于地图的位置。

显然,在上述形式的定位中,机器人的传感器和执行器扮演了必不可少的角色。这是因为这些传感器和执行器的不准确性和不完备性使得定位面临困难的挑战。本节我们确定该传感器和执行器次优性的几个重要方面。

5.2.1 传感器噪声

传感器是感知过程基本的机器人输入。所以,传感器能够区分环境状态的程度是至关重要的。**传感器的噪声**引起了相同环境下对传感器读数一致性的限制,从而,限制了各传感器读数有效的可用位的数目。通常,传感器的噪声问题来源于机器人的表示方法没有抓住某些环境的特征,因而被忽略掉。

例如,在办公楼室内的视觉系统,可能会使用彩色 CCD 摄像机检测到的颜色值。当太阳被云挡住时,由于窗口遍及建筑物,建筑物内部的照明是变化的,因此颜色值不会恒定。如同遭受随机的误差一样,彩色 CCD 摄像机会因机器人的视角而出现噪声。因而从 CCD 摄像机得到的颜色值将是无用的,除非机器人在它的表示中能够注意到太阳和云彩的位置。

照明的相关性只是在基于视觉传感器系统中出现误差的一个例

子。图像抖动、信号增益、模糊现象和图像混乱都是另外的噪声源，它们可能会缩减彩色视频图像可用的内容。

考虑4.1.2.3节所讨论的超声测距传感器(例如声纳)的噪声水平(明显的随机噪声)，当声纳变送器向一个比较平滑且有棱角的表面发射声音时，许多信号会相干地反射掉而不产生回波。虽然根据材料的特性，会有少量的能量返回。当该能量水平接近声纳的增益阈值时，声纳有时会成功，有时却检测失败。从机器人角度看，实际不变的环境状态会产生两种不同的可能读数：一个短，一个长。声纳传感器弱小的信噪比会被多个声纳发射器之间的干扰而进一步变坏。通常，研究的机器人在一个平台上有12～48个声纳。在听觉反射的环境中，一个变送器的声纳发射和另一个变送器回声检测电路之间的多路干扰是可能的。其结果由于一组巧合的角度，在测距值里可能有巨大的误差(即低估计)。这种误差极少发生，小于时间的1%，从机器人角度看，实际上是随机的。

总之，传感器噪声减少了传感器读数有用的信息内容。显然，解决的办法是要考虑到多个读数，运用时间融合和多传感器融合，以增加机器人输入的整个信息内容。

5.2.2 传感器混叠

移动机器人传感器的第二个缺点，是使传感器获得很少的信息内容，并进一步恶化了感知，从而加剧了定位问题。该问题，称之为**传感器混叠**，这是人类很少碰见的现象。人类的传感器系统，特别是视觉系统，倾向于在各唯一的局部状态接收唯一的输入。换句话说，从每一个不同的地方看是不同的。这个唯一映射的能力，只有当一个人在考虑难于把握的情况时才显露出来。让我们考虑一下行走穿过一个完全黑暗且不熟悉的建筑物的情况，当视觉系统只能看见一片黑暗时，人的定位系统就很快退化。另一个有用的例子是由高篱笆做成的人一样大小的迷宫，这种迷宫建造已有好几百年，如果没有

路标或线索,没有视觉的唯一性,人们会发现它们极难求解,人的定位能力也会迅速下降。

机器人传感器读数的非唯一性,或**传感器混叠**,是正常的,不是例外。来看看一个窄束测距仪,如超声或红外测距仪。该传感器在单方向提供距离信息,没有任何有关材料成分,诸如颜色、纹理和硬度的附加数据。即使在一个阵列上具有几个这种传感器的一个机器人,也总存在各种各样的环境状态,它们会跨阵列地触发相同的传感器值。形式上,这是从环境状态到机器人感知输入多对一的映射。因此,机器人的感知不能区分出这许多状态。基于声纳机器人的典型问题,涉及到室内设置中人和非生命物体的区分。当机器人本身面临一个明显的障碍时,机器人是否应该说"对不起请让一下",因为障碍也许是一个走动的人;或者机器人是否应该回绕物体规划一条路径,因为它可能是一个纸板箱。因为仅有一个声纳,所以这些状态是混叠的,不可能区分开来。

传感器混叠给导航带来的问题是:即使用无噪声的传感器,从单个感知读数得来的信息量一般也是不足以识别机器人的位置的。因此,必须通过机器人编程人员运用多种技术,使机器人的定位基于一系列的读数,从而以足够的信息随时知晓机器人的位置。

5.2.3 执行器噪声

定位的难题并不只在传感器技术方面。正像机器人的传感器有噪声,从而限制信号的信息内容一样,机器人的执行器也有噪声。特别是当机器人采取单个动作时,会产生几种不同的可能结果。从机器人的观点看,即使在采取动作之前已清楚知道了初始状态,也会如此。

简言之,机器人执行器引入了有关未来的不确定性。所以,移动的简单动作趋向增加移动机器人的不确定性。当然,这里有例外。利用认知,可以仔细地规划运动,以使这个影响降低到最小。实际上

有些时候确实使结果更加确定。而且，如果机器人采取的动作和传感器反馈的仔细解释相配合，利用传感器所提供的信息可以补偿由噪声动作引入的不确定性。

无论如何，首先要了解影响移动机器人的执行器噪声的实质，这是很重要的。从机器人的观点看，要注意到的是运动的误差被看作是里程表的误差，或者是用它的运动学和动力学知识随时估计它本身位置的不稳定性。误差的真正来源一般来说在于环境的不完整模型。例如，机器人没有对地面可能是斜的、轮子会打滑、人可能推机器人这样的事实建模。所有这些不建模的误差因素，造成了机器人的物理运动、机器人意向运动和本体感受传感器运动估计之间的不准确性。

在里程表（仅对轮子传感器）和航位推测（也对导向传感器）中，位置的更新是基于**本体感受**的传感器。机器人的运动，用轮子编码器或导向传感器或二者读出的数据，集成在一起计算位置。因为集成了传感器测量误差，所以位置误差随时间累加。为此，位置必须用其他定位机制一次次地予以更新。否则，机器人在长时间运行中不能保持有意义的位置估计。

下面，只着重于基于差动驱动机器人的轮子传感器读数的里程表（同样参考[4,57,58]）。使用附加的导向传感器（如陀螺仪）可以帮助减少累计误差，其主要问题是一样的。

里程表有多种误差源，从环境因素到分辨率：

- 综合期间有限的分辨率（时间增量、测量分辨率等）；
- 轮子的不正确配置（确定性的）；
- 轮子直径的不确定性，特别是轮子直径不等（确定性的）；
- 轮子接触点的变动；
- 不等的地面接触（斜坡、非平面等）。

有些误差可能是**确定性的（系统性的）**，因此可以用系统的适当校正方法予以消除。然而，仍然存在许多**不确定的（随机的）**误差，随

时导致位置估计的不确定性。从几何观点看，可将误差分成三种类型：

(1) 距离误差：机器人运动的整个路径长度(距离)→轮子移动的总和；

(2) 转动误差：类似于距离误差，但是因转动产生→轮子运动之差；

(3) 漂移误差：轮子误差和差异，导致机器人角度方向误差。

经过长的时间周期，转动和漂移误差远远大于距离误差，因为它们对整个位置误差的影响是非线性的。考虑一个其初始位置完全知道的机器人，沿着 x 轴在一条直线上向前移动，移动 d 米后，在 y 位置所引进的误差为 $d\sin\Delta\theta$ 分量。当误差角 $\Delta\theta$ 增大时，这分量可能会很大。整个时间内，当移动机器人在环境周围运动时，其内部参考框架和原始参考框架之间的旋转误差快速地增加。随着机器人从这些框架的原点移开，最终位置的线性误差增加很大。所以，为了计程准确，建立误差模型以了解误差在整个时间内如何传播是有好处的。

5.2.4 里程表位置估计的误差模型

一般而言，机器人的姿态(位置)可以用以下向量表示：

$$p=\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{5.1}$$

对于差动驱动的机器人来说(图 5.3)，其位置可以从一个已知位置开始，并将运动进行积分(行走距离的增量求和)予以估计。对具有固定采样间隔 Δt 的离散系统，行走距离的增量(Δx；Δy；$\Delta\theta$)为：

$$\Delta x=\Delta s\cos(\theta+\Delta\theta/2) \tag{5.2}$$

$$\Delta y=\Delta s\sin(\theta+\Delta\theta/2) \tag{5.3}$$

$$\Delta\theta=\frac{\Delta s_r-\Delta s_l}{b} \tag{5.4}$$

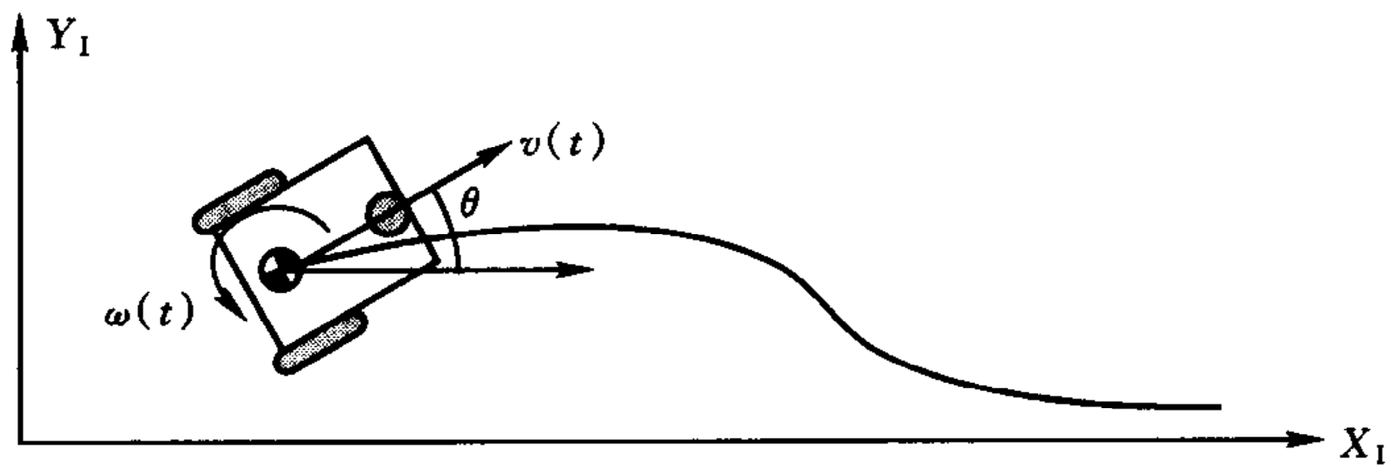


图 5.3 差动驱动机器人的运动

$$\Delta s=\frac{\Delta s_r+\Delta s_l}{2} \tag{5.5}$$

式中：$(\Delta x;\Delta y;\Delta\theta)$＝前一次采样间隔走过的路径；

$\Delta s_r;\Delta s_l$＝分别为左右轮行走的距离；

b＝差动驱动机器人两个轮子之间的距离。

由此，我们得到更新过的位置 p'：

$$p'=\begin{bmatrix}x'\\y'\\\theta'\end{bmatrix}=p+\begin{bmatrix}\Delta s\cos(\theta+\Delta\theta/2)\\\Delta s\sin(\theta+\Delta\theta/2)\\\Delta\theta\end{bmatrix}=\begin{bmatrix}x\\y\\\theta\end{bmatrix}+\begin{bmatrix}\Delta s\cos(\theta+\Delta\theta/2)\\\Delta s\sin(\theta+\Delta\theta/2)\\\Delta\theta\end{bmatrix} \tag{5.6}$$

利用方程(5.4)和(5.5)的$(\Delta s;\Delta\theta)$关系，我们进一步得到里程表位置更新的基本方程(对差动驱动机器人)：

$$p'=f(x,y,\theta,\Delta s_r,\Delta s_l)=\begin{bmatrix}x\\y\\\theta\end{bmatrix}+\begin{bmatrix}\frac{\Delta s_r+\Delta s_l}{2}\cos\left(\theta+\frac{\Delta s_r-\Delta s_l}{2b}\right)\\\frac{\Delta s_r+\Delta s_l}{2}\sin\left(\theta+\frac{\Delta s_r-\Delta s_l}{2b}\right)\\\frac{\Delta s_r-\Delta s_l}{b}\end{bmatrix} \tag{5.7}$$

如我们以前讨论过那样，里程表位置更新只能给出实际位置的一个非常粗略的估计。由于在增量运动$(\Delta s_r;\Delta s_l)$期间，p 的不确定性积分误差和运动误差，基于里程表积分的位置误差随时间而增加。

下一步，我们将建立整体位置 p'的误差模型，得到里程表位置

估计的协方差矩阵 $\Sigma_{p'}$。为此,我们假定在起始点初始协方差矩阵 Σ_p 为已知。对运动增量(Δs_r;Δs_l),我们假定协方差矩阵 Σ_Δ 为:

$$\Sigma_\Delta = covar(\Delta s_r;\Delta s_l) = \begin{bmatrix} k_r|\Delta s_r| & 0 \\ 0 & k_l|\Delta s_l| \end{bmatrix} \tag{5.8}$$

式中 Δs_r 和 Δs_l 是各轮行走的距离,k_r 和 k_l 是误差常数,代表马达驱动和轮子-地面交互的非确定性参数。可以看到,在方程(5.8)中我们做了以下的假设:

- 各个驱动轮的两个误差是独立的*;
- 误差(左轮和右轮)的方差正比于行走过的距离(Δs_r;Δs_l)的绝对值。

这些假定虽然不理想,但是合适的。因此将被用于误差模型的进一步开发。运动误差是因为轮子变形、滑动、地面不平、编码器误差等造成不精确运动所致。误差常数 k_r 和 k_l 的值取决于机器人和环境,应该通过执行和分析有代表性的运动由实验确定。

如果我们假定 p 和 $\Delta_{rl}=(\Delta s_r;\Delta s_l)$不相关,$f$ 的微分(方程(5.7))由一阶泰勒展开(线性化)合理地近似,利用误差传播定律(见 4.2.2 节),我们得出结论:

$$\Sigma_{p'} = \nabla_p f \Sigma_p \nabla_p f^T + \nabla_{\Delta_{rl}} f \Sigma_\Delta \nabla_{\Delta_{rl}} f^T \tag{5.9}$$

当然,协方差矩阵 Σ_p 常常由前一步 $\Sigma_{p'}$ 给定,因而在指定一个初始值(譬如,0)后,就可以计算。

利用方程(5.7),我们可以计算两个**雅可比**,$F_p=\nabla_p f$ 和 $F_{\nabla rl}=\nabla_{\Delta_{rl}} f$:

$$F_p = \nabla_p f = \nabla_p(f^T) = \left[\frac{\partial f}{\partial x}\ \frac{\partial f}{\partial y}\ \frac{\partial f}{\partial \theta}\right] = \begin{bmatrix} 1 & 0 & -\Delta s\sin(\theta+\Delta\theta/2) \\ 0 & 1 & \Delta s\cos(\theta+\Delta\theta/2) \\ 0 & 0 & 1 \end{bmatrix} \tag{5.10}$$

* 如果有更多的关于实际机器人的运动学知识,也可以用协方差的相关项。

$$F_{\Delta_{rl}}=\begin{bmatrix}\frac{1}{2}\cos\left(\theta+\frac{\Delta\theta}{2}\right)-\frac{\Delta s}{2b}\sin\left(\theta+\frac{\Delta\theta}{2}\right) & \frac{1}{2}\cos\left(\theta+\frac{\Delta\theta}{2}\right)+\frac{\Delta s}{2b}\sin\left(\theta+\frac{\Delta\theta}{2}\right)\\ \frac{1}{2}\sin\left(\theta+\frac{\Delta\theta}{2}\right)+\frac{\Delta s}{2b}\cos\left(\theta+\frac{\Delta\theta}{2}\right) & \frac{1}{2}\sin\left(\theta+\frac{\Delta\theta}{2}\right)-\frac{\Delta s}{2b}\cos\left(\theta+\frac{\Delta\theta}{2}\right)\\ \frac{1}{b} & -\frac{1}{b}\end{bmatrix} \tag{5.11}$$

方程(5.11)的细节是：

$$F_{\Delta_{rl}}=\nabla_{\Delta_{rl}}f=\left[\frac{\partial f}{\partial\Delta s_r}\ \frac{\partial f}{\partial\Delta s_l}\right]=\cdots \tag{5.12}$$

$$\begin{bmatrix}\frac{\partial\Delta s}{\partial\Delta s_r}\cos\left(\theta+\frac{\Delta\theta}{2}\right)+\frac{\Delta s}{2}-\sin\left(\theta+\frac{\Delta\theta}{2}\right)\frac{\partial\Delta\theta}{\partial\Delta s_r} & \frac{\partial\Delta s}{\partial\Delta s_l}\cos\left(\theta+\frac{\Delta\theta}{2}\right)+\frac{\Delta s}{2}-\sin\left(\theta+\frac{\Delta\theta}{2}\frac{\partial\Delta\theta}{\partial\Delta s_l}\right)\\ \frac{\partial\Delta s}{\partial\Delta s_r}\sin\left(\theta+\frac{\Delta\theta}{2}\right)+\frac{\Delta s}{2}\cos\left(\theta+\frac{\Delta\theta}{2}\right)\frac{\partial\Delta\theta}{\partial\Delta s_r} & \frac{\partial\Delta s}{\partial\Delta s_l}\sin\left(\theta+\frac{\Delta\theta}{2}\right)+\frac{\Delta s}{2}\cos\left(\theta+\frac{\Delta\theta}{2}\right)\frac{\partial\Delta\theta}{\partial\Delta s_l}\\ \frac{\partial\Delta\theta}{\partial\Delta s_r} & \frac{\partial\Delta\theta}{\partial\Delta s_l}\end{bmatrix} \tag{5.13}$$

用方程(5.14)和(5.15)，我们得到方程(5.11)。

$$\Delta s=\frac{\Delta s_r+\Delta s_l}{2};\quad \Delta\theta=\frac{\Delta s_r-\Delta s_l}{b} \tag{5.14}$$

$$\frac{\partial\Delta s}{\partial\Delta s_r}=\frac{1}{2};\ \frac{\partial\Delta s}{\partial\Delta s_l}=\frac{1}{2};\ \frac{\partial\Delta\theta}{\partial\Delta s_r}=\frac{1}{b};\ \frac{\partial\Delta\theta}{\partial\Delta s_l}=-\frac{1}{b} \tag{5.15}$$

图 5.4 和图 5.5 表示了位置误差如何随时间增加的典型例子。利用上面提出的误差模型计算出了结果。

一旦建立了误差模型，必须指定误差参数。人们可以恰当地校正机器人以补偿确定性误差。然而，指定非确定性误差的参数只能由统计(重复的)测量进行量化。里程误差的详细讨论，确定性和非确定性误差的校正和量化方法参阅文献[5]。在文献[105]提出了一种迅速的里程误差估计方法。

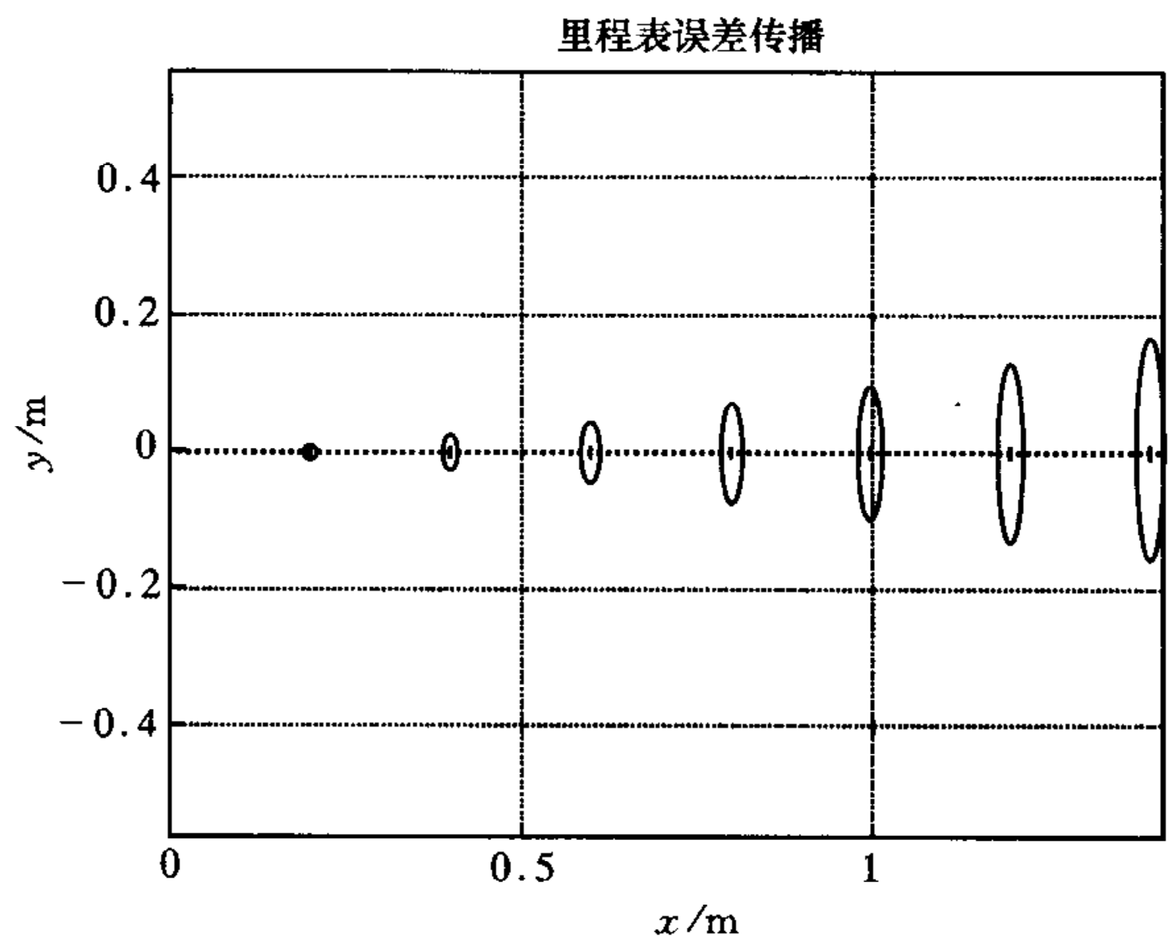


图 5.4 直线运动姿态不确定性的增长:注意到 y 不确定性增长比运动方向更快。这结果是由于不确定性对机器人方位积分造成的。环绕机器人位置所画的椭圆代表在 x,y 方向(例如,3σ)的不确定性。图中没有表示 θ 方位的不确定性,虽然它的影响可以间接观测到

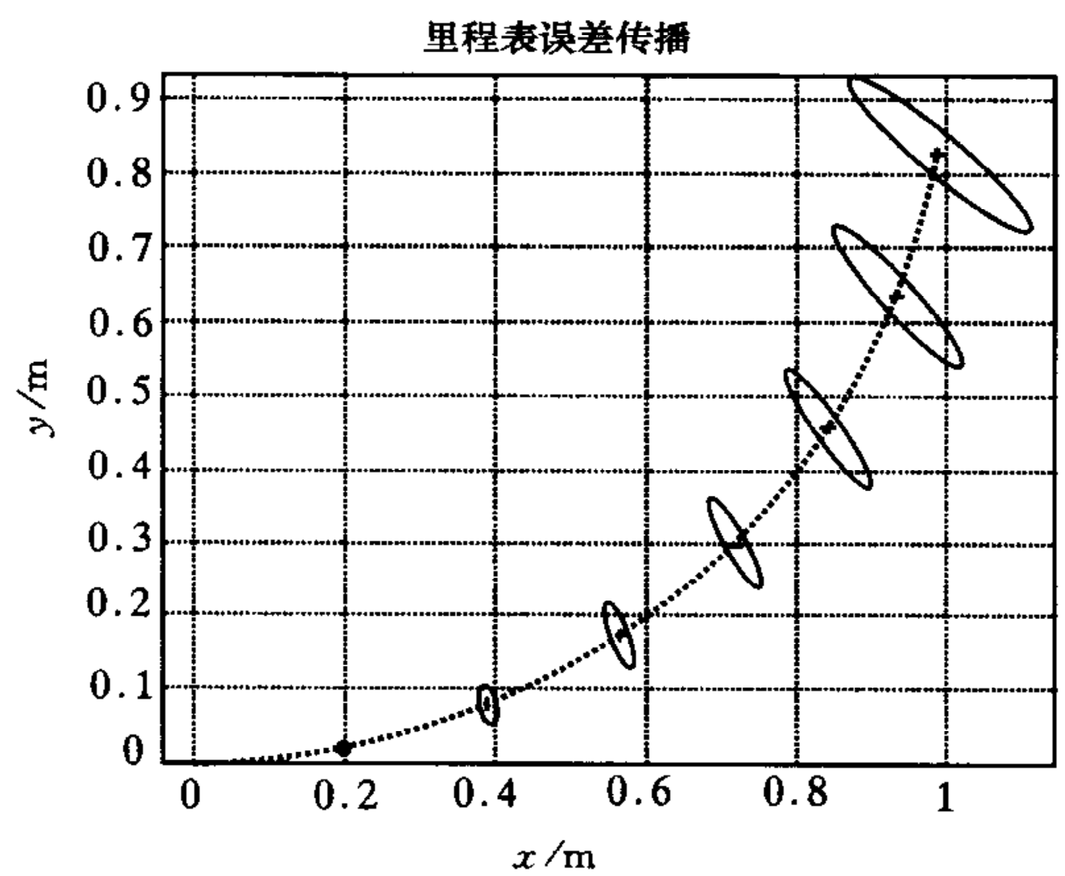


图 5.5 圆周运动(r=常数)姿态不确定性的增长:再次,垂直于运动的不确定性比运动方向的增长快得多。注意到不确定性椭圆的主轴与运动方向不保持垂直

5.3 定位或不定位:基于定位的导航与编程求解的对比

图5.6描述了移动机器人导航的标准室内环境。假定要讨论的移动机器人必须在这环境中的两个特定房间A和B之间传送消息。在建立导航系统中,移动机器人显然需要传感器和移动控制系统。绝对地要求传感器避免碰撞运动的障碍物,诸如人,要求某些运动控制系统能使机器人谨慎地移动。

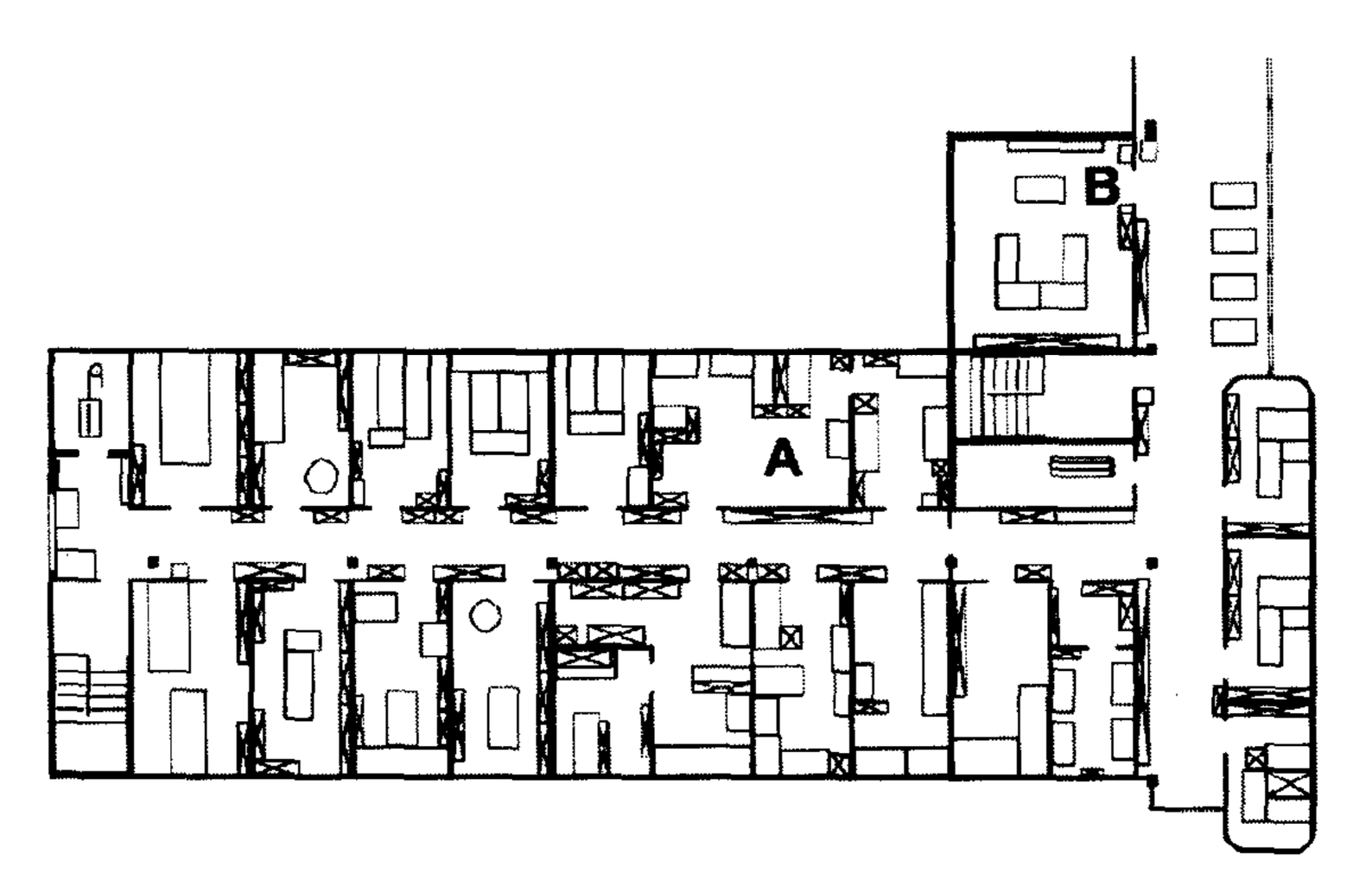


图5.6 实例环境

然而,该机器人是否会要求一个**定位系统**还缺少证明。为了在两个房间之间成功地导航,定位似乎是强制性的。毕竟,通过在地图上定位,机器人可以期望恢复它的位置,并当它已经到达目标位置时进行检测。确实,机器人至少必须有一个检测目标位置的方法。然而,参考地图明显地定位,不是有条件成为目标检测器的唯一策略。

另一种方法受到基于行为的团体的支持,他们认为不应该建立定位的几何地图,因为传感器和执行器有噪声且信息是有限的。该团体建议代之以设计合在一起的行为集合,形成期望的机器人运动。

从根本上说,该方法消除了对定位和位置的显式推理。因而一般地说,它同样取消了显式的路径规划。

该技术基于一个信念,即对当前的特殊导航问题有一个程序上的解。例如,在图 5.6 中,从房间 A 到房间 B,行为主义者的导航方法也许是设计一个左墙壁的跟踪行为和房间 B 的检测器,该检测器由房间 B 中某独特的队列所触发,诸如地毯的颜色。然后利用左墙壁跟踪器与房间检测器的交,合作成为程序结束的条件,机器人可以到达房间 B。

对一个特殊的导航问题,该求解方案的体系结构如图 5.7 所示。这个方法的主要优点是:对具有少数目标位置的单独环境,可能很快可以实现。然而它也有某些缺点。首先,这方法不可推广到其他的环境或较大的环境中。通常,导航程序是按位置专用的,将机器人移到新的环境需要同样程度的编程和诊断。

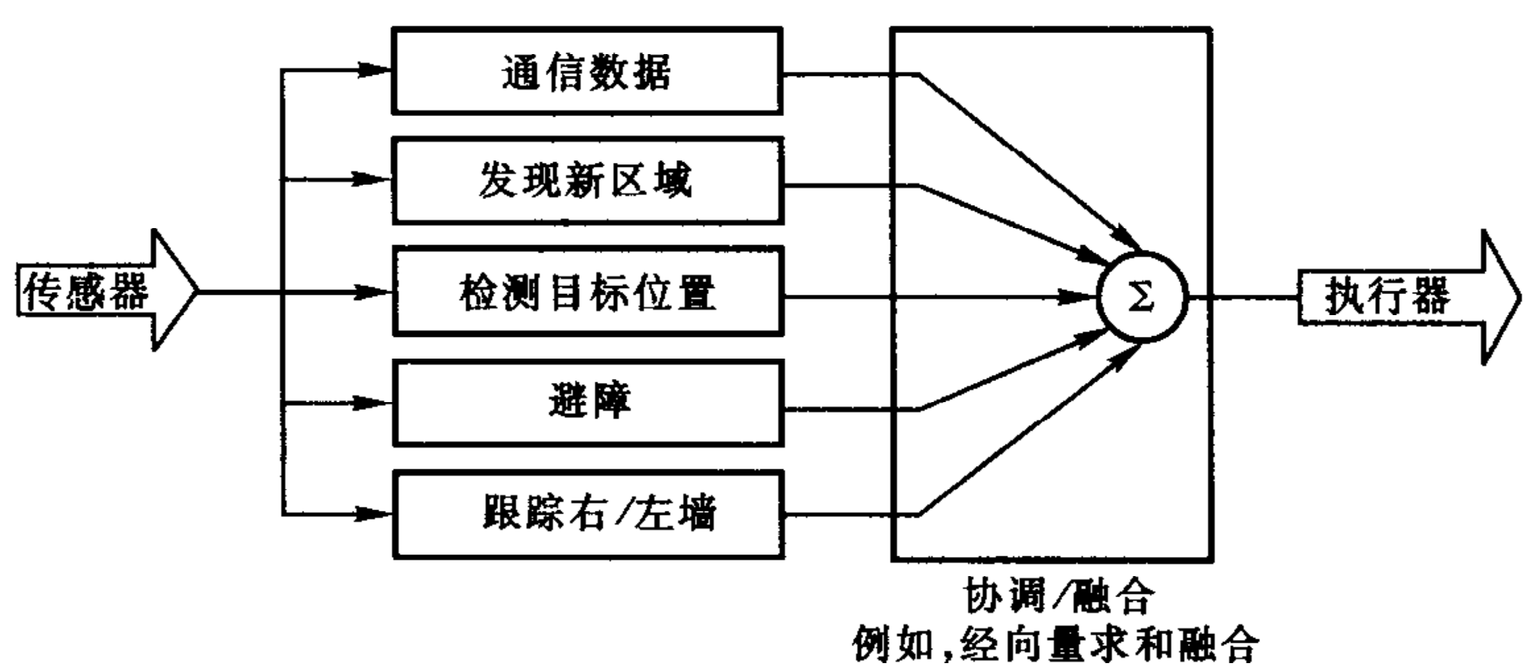


图 5.7 基于行为导航的结构

其次,基本程序,诸如**跟踪左边墙**,必须仔细设计以产生期望的行为。这任务可能是费时的,而且严重地依赖于特定机器人的硬件和环境特征。

第三,在任何一个时刻,基于行为的系统需有多个主动行为,甚至当单个行为被调整到最优性能时,多个行为之间的融合和快速切换可以使那个精密的调节无效。各个新增行为的加入,常常迫使机

器人的设计者再次全部退回到现有的行为，以保证与新引入的行为的新交互全都稳定。

与基于行为的方法相反，基于地图的方法包括了**定位和认知**模块(图5.8)。在基于地图的导航中，机器人明确地力图通过收集传感器的数据，随后相对于环境图更新有关它位置的某些信任度进行定位。基于地图的导航方法，其主要优点如下：

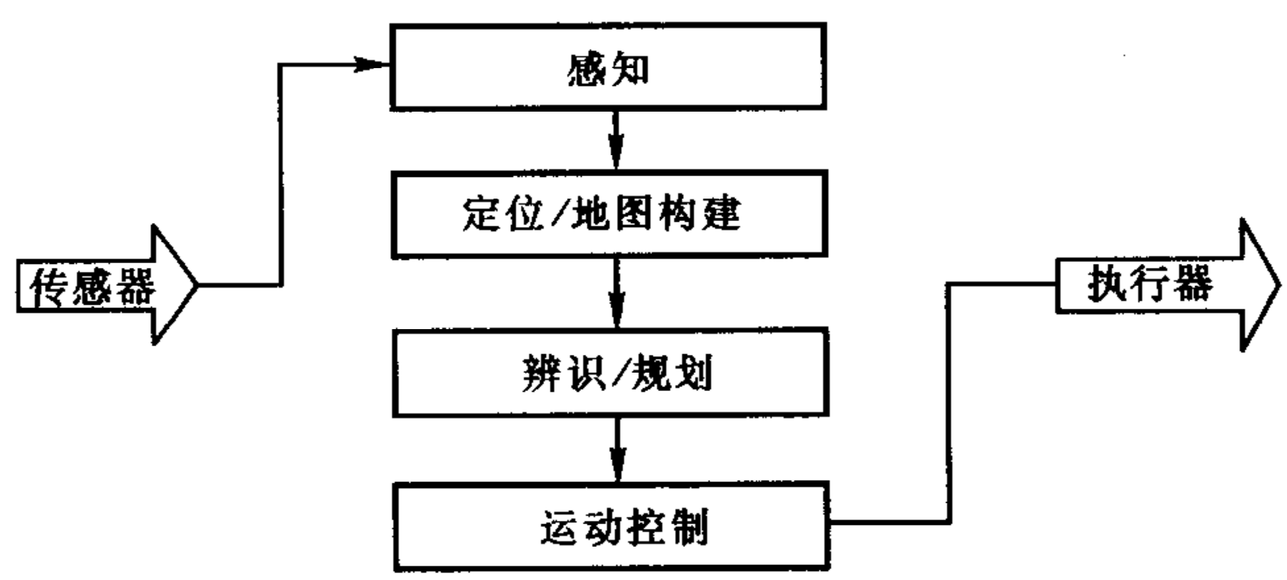


图5.8 基于地图(基于模型)导航的结构

- 显式的、基于地图的位置概念，使得有关位置的系统信任度对人类操作员透明可用；
- 地图存在的本身代表了人与机器人之间交流的一个媒介：如果机器人走到一个新的环境，人可以简单地给机器人一张新的地图；
- 如果机器人建立了地图，人同样可以利用，达到双用的目的。

基于地图的方法会需要更前沿的开发成果，以创建导航的移动机器人。希望开发的成果产生可以成功地作图和导航各种不同环境的体系结构。由此，随着时间的推移，分期回报超前的设计耗费。

当然，基于地图的方法的主要风险是机器人所构造和**信赖**的内部的表示，而不是现实世界本身。如果模型偏离现实(即，地图是错的)，即使机器人的原本传感器的值仅仅是传输上不正确，那么机器人的行为也会不符合要求。

在本章的其余部分，我们集中讨论基于地图的方法。而且特别

集中讨论这些技术的定位组成部分。如果意义重大的最新成果能使移动机器人导航各种不同环境，从学术研究大楼到工厂车间，到世界各地的博物馆，那么这些方法就特别适合于研究。

5.4 信任度的表示

区分不同的基于地图的定位系统，其最基本的问题是**表示**。有两个机器人必须表达且各有它自己独特的可能解的特殊概念：机器人必须有一个环境的表示方法（一个模型）或一张地图。在地图中包含环境哪些方面？地图以何种精确度表示环境？这些就是**地图表示方法**的设计问题。

机器人还必须有一个关于在地图上表示它的位置信任度的方法。机器人是否辨识一个单独的特定位置作为它的当前位置，或它是否根据一组可能的位置描述它的位置？如果以单个信任度完全表达多个可能位置，那如何将这多个位置排序？这些就是**信任度表示方法**的设计问题。

沿着这两个设计轴，决策可以形成不同水平的结构复杂性、计算复杂性和整体定位准确度。我们以讨论信任度的表示方法开始。在信任度表示系统的术语中，最重要的分支是区分单假设和多假设信任度系统之间的差别。前者包含着机器人假定其唯一位置的解答；而后者能使机器人描述有关它位置的不确定度。不同的信任度和地图表示的实例如图 5.9 所示。

5.4.1 单假设信任度

单假设信任度表示是移动机器人位置最直接可能的假设。给定某个环境地图，机器人关于位置的信任度表示为地图上单个独特的点。在图 5.10 表示了相同环境（图 5.10(a)）中，用三种不同地图表

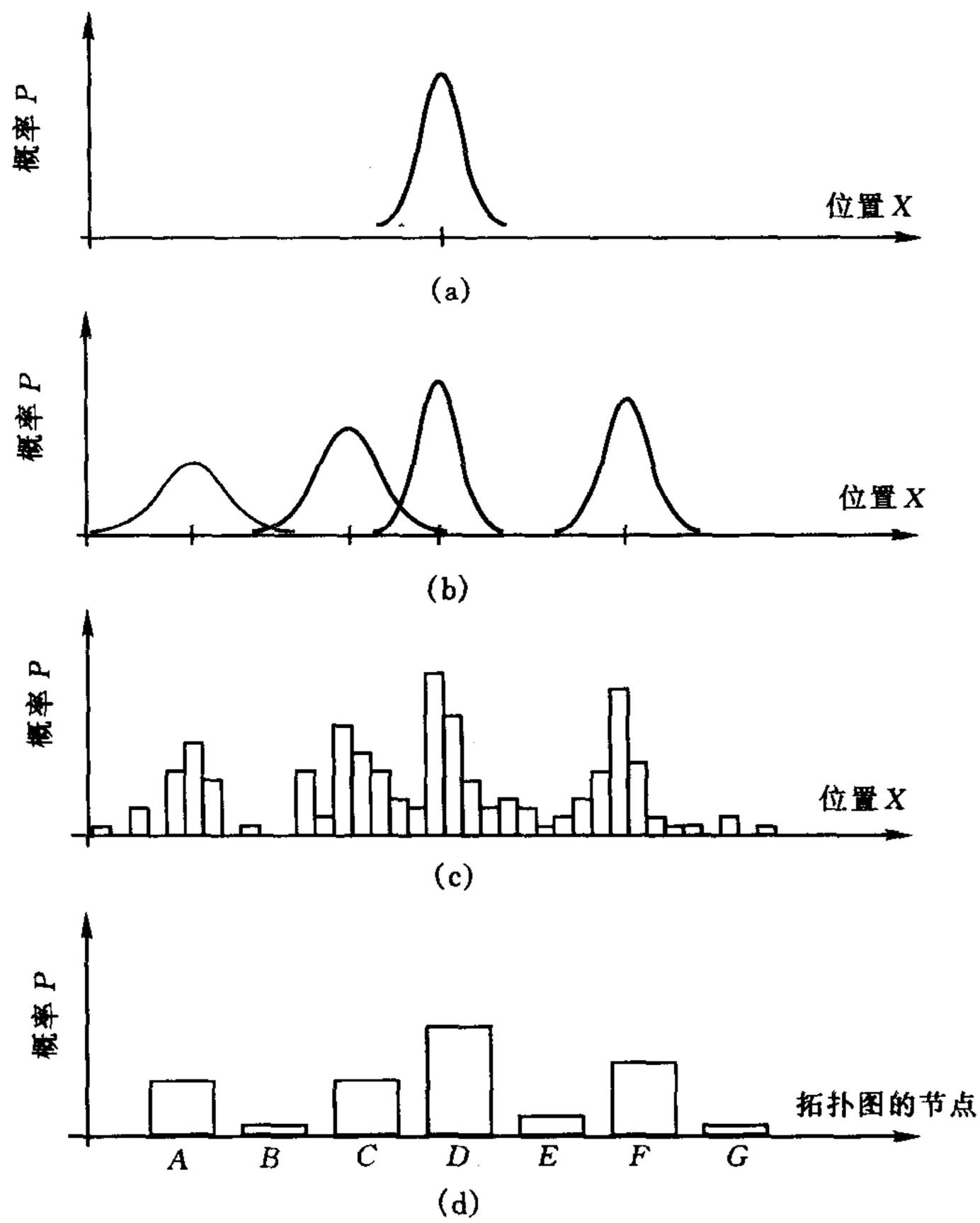


图 5.9 在连续和离散(棋盘栅格)地图中关于机器人位置(1D)信任度表示。(a)具有单假设信任度的连续图,例如中心在单连续值的单个高斯函数;(b)具有多假设信任度的连续图,例如中心在多连续值的多高斯函数;(c)对所有可能的机器人位置,具有概率值离散(分解)栅格图,例如马尔可夫方法;(d)对所有可能节点(拓扑上的机器人的位置),具有概率值的离散拓扑图,例如马尔可夫方法

示方法的单假设信任度的三个例子。在图 5.10(b),单点几何上标注是连续 2D 几何图的机器人位置。在图 5.10(c),地图是离散、棋盘栅格图,位置标注的保真度与地图单元尺寸保真度相同。在图 5.10(d),地图根本不是几何形状,而是抽象和拓扑的。在这种情况下,位置的单个假设,包含将拓扑图中单个节点 i 认定为机器人的位

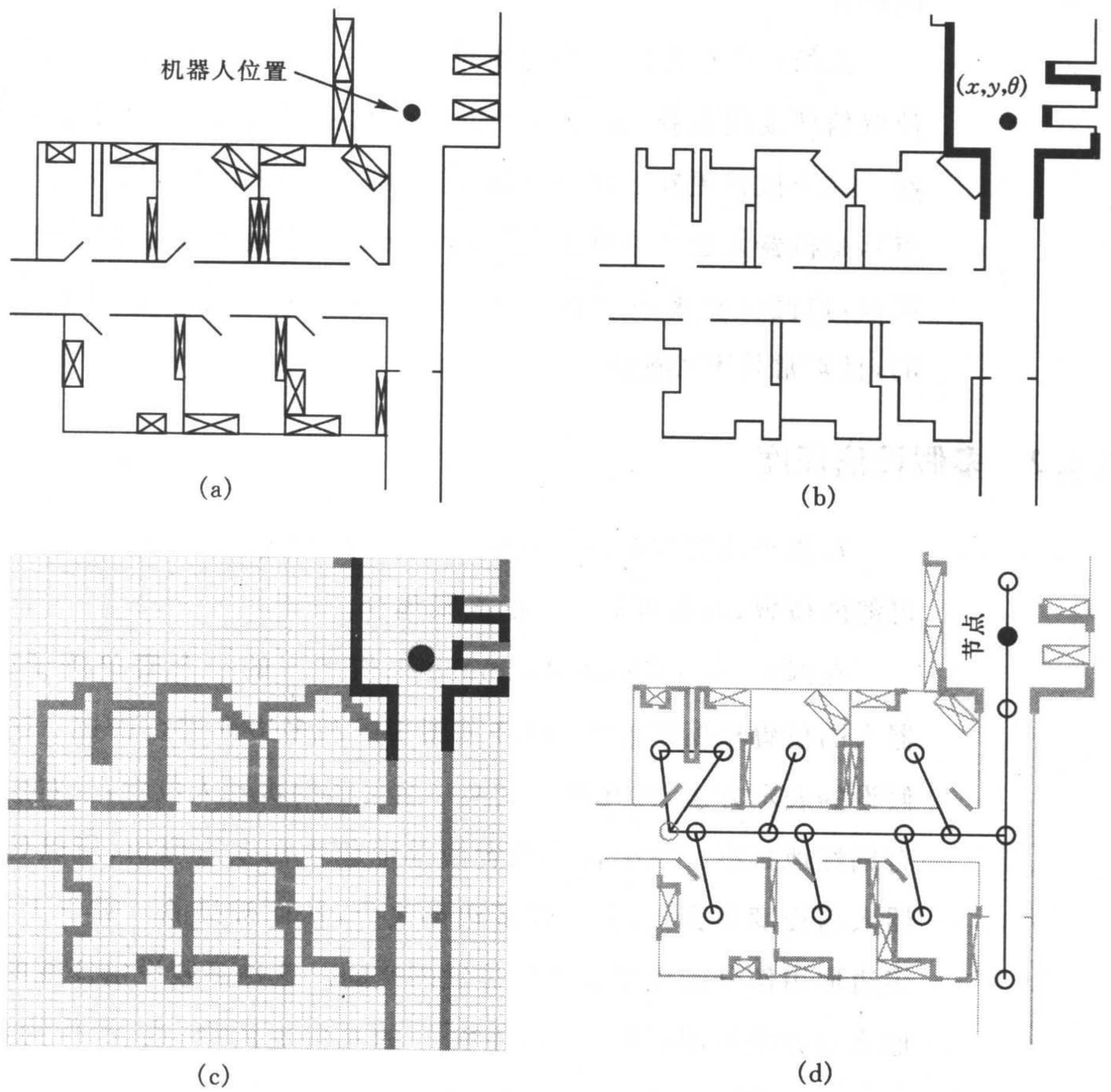


图 5.10 用不同的地图表示方法，位置单个假设的三个例子：(a)具有墙、门和家具的真实地图；(b)基于直线的图→有 2 个参数，约 100 条线；(c)基于占有栅格的地图→栅格单元尺寸为 50 cm×50 cm，约 3000 个栅格；(d)用线特征(Z/S)和门的拓扑图→约 50 个特征和 18 个节点

置。从事实可知，位置的单假设表示方法，其主要优点是：给定唯一的信任度，位置没有任意性。这个表示方法的确定性实质加速了机器人在认知水平上(例如，路径规划)的决策。机器人可以简单地假定，它的信任度是正确的，然后可以根据它唯一的位置，选择其未来

的动作。

正像单个位置假设加速决策过程一样，它也加速了机器人关于位置信任度的更新，因为单个位置必须被新定义的、单个位置所更新。这个位置更新方法的困难（它最终成为单个假设表示的主要缺点），是机器人运动常常因执行器和传感器的噪声而引起不确定性。所以，迫使位置更新过程总是产生位置的单个假设，是具有挑战性的，且常常是不可能的。

5.4.2 多假设信任度

在关于位置多假设信任度的情况下，机器人跟踪不仅仅是单个可能的位置，而是可能位置的无限集合。

在源于 Jean-Claude-Latombe 工作[21,99]的一个简单例子中，机器人的位置根据定位于 2D 环境图中的凸多边形进行描述。这个多假设的表示方法，在几何上交换一组可能的机器人位置，且对位置没有偏好地排序。图中各点或者简单地包含在多边形中，所以处在机器人信任度集合内；或者在多边形之外，从而被排除。数学上，位置多边形用作分割机器人可能位置的空间。这种多假设信任度的多边形表示方法可用于连续的、环境[35]的几何地图中；或者，作为选择，用于棋盘格形、连续环境的离散逼近中。

然而，注意到某些机器人的位置比其他位置更可能，将可能的机器人位置具体表达为某种排序，也许是有用的。与可能位置偏好排序结合在一起，表示连续多假设信任状态的策略是：将信任度建模成一个数学分布。例如，参考文献[50,142]中利用 2D 环境中的点$\{X, Y\}$，把机器人位置信任度标记为平均μ加上一个标准偏差参数σ，由此定义了一个高斯分布。意向的解释是：各位置的分布代表了分配给机器人在该位置的概率。这个表示方法特别适从于数学上所定义的跟踪函数，诸如卡尔曼滤波器，它被设计成在高斯分布上有效的运算。

另一个表示可能机器人位置集合的办法，不用单个高斯概率密度函数，而是各可能位置的离散标记。在这种情况下，各个可能的机器人位置，各自地与自信度或概率参数结合在一起被标记(图5.11)。在高度棋盘格化的地图中，这可在单信任度状态中，造成成千或几万个可能的机器人位置。

图 5.11 多假设跟踪(取自 W. Burgard)的例子。移动到位置 4 以后，大量的分布的信任度状态成为十分确定

多假设表示方法的主要优点是：机器人可以明确地维持关于它位置的不确定性。如果机器人只从它的传感器和执行器获取部分的关于位置信息，则从概念上该信息可以被合并成为更新的信任度。这个方法更为稀有的优点是：它是以机器人明确地测量它自己关于位置不确定性程度的能力为中心。这个优点对一类定位和导航求解是关键的。在这类问题中，机器人不仅对达到特定目标要推理，还要对它自己信任度状态的未来轨迹进行推理。譬如，机器人也许要选择使它的未来位置不确定性最小的路径。该方法的一个例子是参考文献[141]。在该例子中，机器人规划一条从点 A 到点 B 的路径，为了减轻定位困难，它采取一系列与路标接近的路径。这类关于轨迹对定位质量有影响的显式推理，就需要多假设表示方法。

多假设方法的主要缺点之一涉及到决策过程，如果机器人把它的位置表示成可能位置的一个区域或集合，那么它如何决定下一步该做什么？图 5.11 提供了一个实例。在位置 3，机器人的信任度状

态分别分布在5个走廊中间，如果机器人的目标是行走到一个特殊走廊，那么给定该信任度状态，机器人应该选什么动作？

因为机器人的某些可能位置隐含着一条与其他某些可能位置不一致的运动轨迹。我们在下面实例研究中将会看到，为决策起见，有一个方法是假定在机器人的信任度状态中，机器人处在物理上最可能的位置，然后根据当前的位置选择一条路径。但是，这个方法要求各可能位置有一个相关的概率。

一般而言，对这种决策问题，正确的方法是根据轨迹进行决策，以消除明显的模糊性。但这导致多假设方法的第二个主要缺点：在绝大多数情况下，它们在计算上十分昂贵。当一个人在离散可能位置的3D空间中推理时，在单假设情况下，可能的信任度状态的数目限制在3D世界可能位置数目内。设此数目为N，当他移动到任意的多假设表示时，那么可能的信任度状态的数目是N的幂集，远远大于2^N。因此，整个时间内关于信任度状态的可能轨迹，其显式推理随着环境规模的增大，很快变得计算上不可容忍。

然而，存在许多稍微多一些约束的多假设表示方法的特殊形式。它允许有有限类型的多假设信任度，同时可避免计算上的爆炸。例如，如果假定概率的高斯分布的中心处在单个位置上，那么表示和信任度跟踪问题变成等效于卡尔曼滤波，即下述的简单的数学过程。另一种方法，在信任度状态中，与有限的10个可能位置结合一起的高度棋盘格化的地图表示，形成离散的更新循环。它最坏只比单假设信任度更新多十倍计算耗费。其他与复杂性问题相适配，而又精确和计算便宜的方法是混合度量-拓扑方法[145,147]或多高斯位置估计[35,60,81]。

总之，多假设信任度状态的最重要优点是持有对位置敏感的能力，同时明确地标注关于机器人它自己位置的不确定性。这个强有力的表示方法，正如在下面实例研究将看到的那样，已经使机器人能以有限的传感器信息在环境的格局中鲁棒地导航。

5.5 地图表示方法

表示机器人移动的环境问题，是表示机器人可能位置或位置的对偶问题。有关环境表示方法所做的决策，会影响到机器人位置表示可用的选择。通常，位置表示的准确性受到地图表示准确性的限制。

在选择一个特殊的地图表示方法时，必须了解三个基本关系：

(1) 地图的精度必须恰当地匹配机器人需要达到目标的精度；

(2) 地图的精度和所表示的特征类型必须匹配机器人传感器所返回的数据类型；

(3) 地图表示的复杂性直接影响有关作图、定位和导航推理的计算复杂性。

在以下章节，我们确定和讨论建立地图表示方法中关键的设计选择。各种选择极大地影响上述所列的关系和最终机器人定位的体系结构。我们将会看到，可能的地图表示方法的选择范围是广泛的。选择一个恰当的表示方法，需要了解在该选择中内在的所有折衷，以及了解一个特别的移动机器人必须履行定位的特殊背景。一般来说，环境表示和模型可以像第 4 章 4.3 节所提出的那样，粗糙地进行分类。

5.5.1 连续的表示方法

连续值地图是**环境**精确分解的一种方法。环境特征的位置可以在连续空间中精密地予以标记。当前实现的移动机器人，只在 2D 表示中使用连续地图，维数增大会引起计算上的爆炸。

一个通用的方法是将连续表示的精确性和**闭环世界假设的紧凑性**结合起来。这意味着，人们设想表示会在地图中指定所有的环境

物体，地图中没有物体的任何区域，在相应的环境部分也就没有物体。因此，地图中所需的总存储量正比于环境中物体密度，而稀疏的环境可由低存储的地图来表示。

这种表示的一个例子，如图5.12所示，是一个2D表示方法。这里，多边形表示连续值坐标空间中所有障碍物。这类似于Latombe[21,98]所用的方法和用于移动机器人路径规划技术的其他环境表示方法。

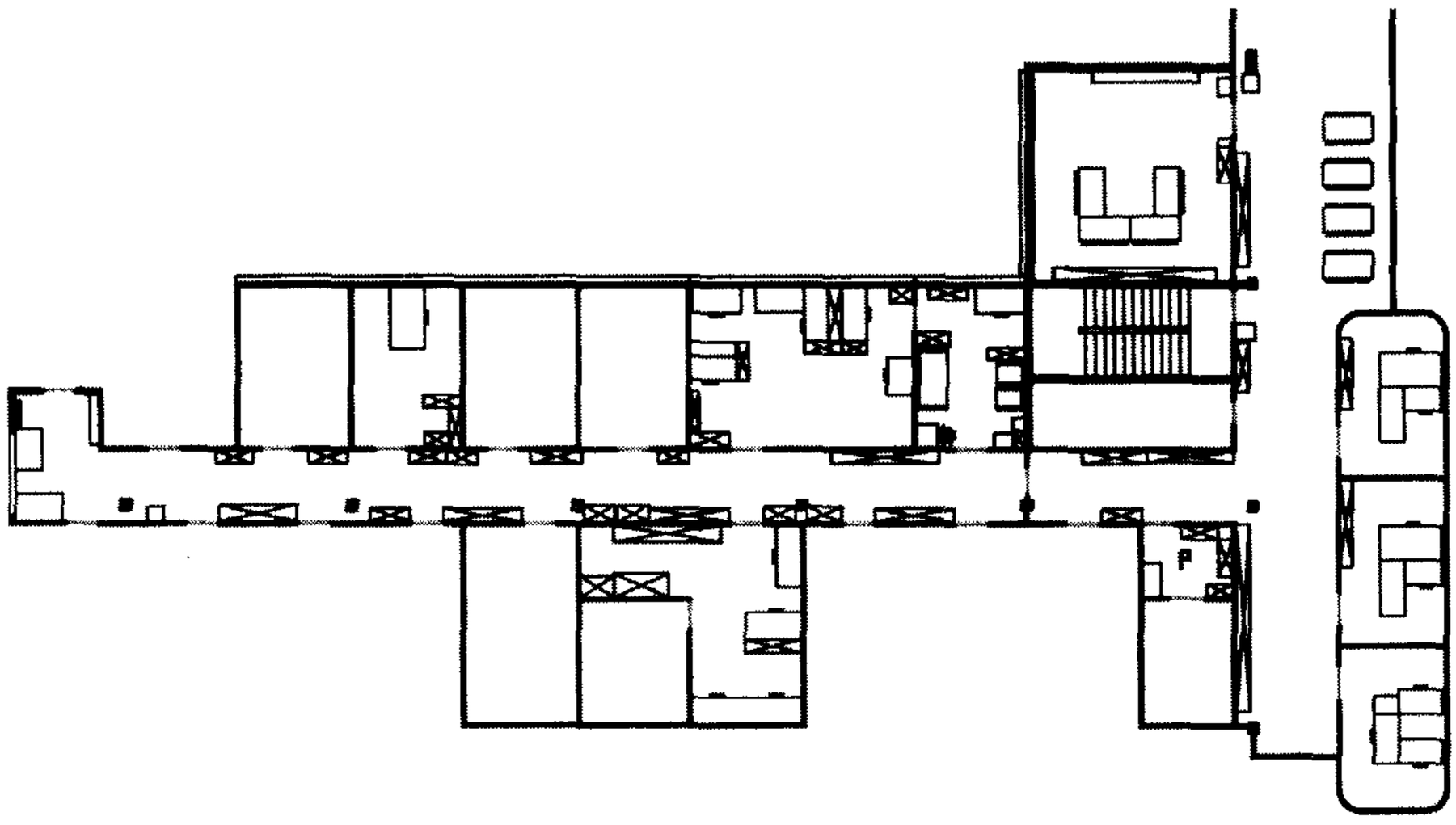


图5.12 用多边形作环境障碍物的连续表示方法

在文献[21,98]中的情况下，事实上大多数的实验无一例外地是在计算机存储器内运行仿真。所以，实际成果没有被推广到用多边形集合去描述真实世界的环境，诸如停车场或办公大楼。

在必须由地图捕获现实环境的其他研究工作中，人们看到了向选择性和抽象性发展的趋向。为定位起见，人为地图的创作者倾向于只在地图上捕捉能被机器人传感器检测到的对象，而且只是现实世界物体特征的一个子集。

我们立刻应该明白，几何图形有能力表示物体的物理位置，不必提及它们的纹理、颜色、弹性，或任何其他不直接关系到位置和空间

的二次特征。除了这个层次的简化之外，通过只捕捉与定位紧密相关的物体几何学方面特征，移动机器人的地图可进一步精简存储器的使用。例如，所有物体可以用非常简单的多边形近似。为了保证计算速度，会牺牲地图的巧妙性。

一个精彩的例子涉及直线的提取。许多室内机器人依赖激光测距装置发现附近机器人的距离数据，这种机器人可以从成千上万激光测点所提供的密集的距离数据中自动地提取最佳拟合的直线。给定这种直线提取传感器，一个合适的连续作图方法就是用无限的直线集合构造地图。地图的连续本质保证可将直线置于平面中任何位置，任何角度。现实环境物体（诸如墙和交叉口）的抽象，只捕捉在地图表示中与移动机器人测距传感器所发现的信息类型相匹配的信息。

图 5.13 展示了利用连续直线表示方法的 EPFL 室内环境的地图。注意到地图捕捉唯一的环境特征是直线，诸如在角落和沿墙的直线。这不仅表示了特征较为丰富的现实世界的样本，而且也是一种简化。对于实际的墙来说，它可以有未被所画直线所捕获的纹理和浮雕。

连续地图表示方法对位置表示的影响基本上是正面的。在单假设位置表示的情况下，位置可以被指定为坐标空间中的连续值的点，

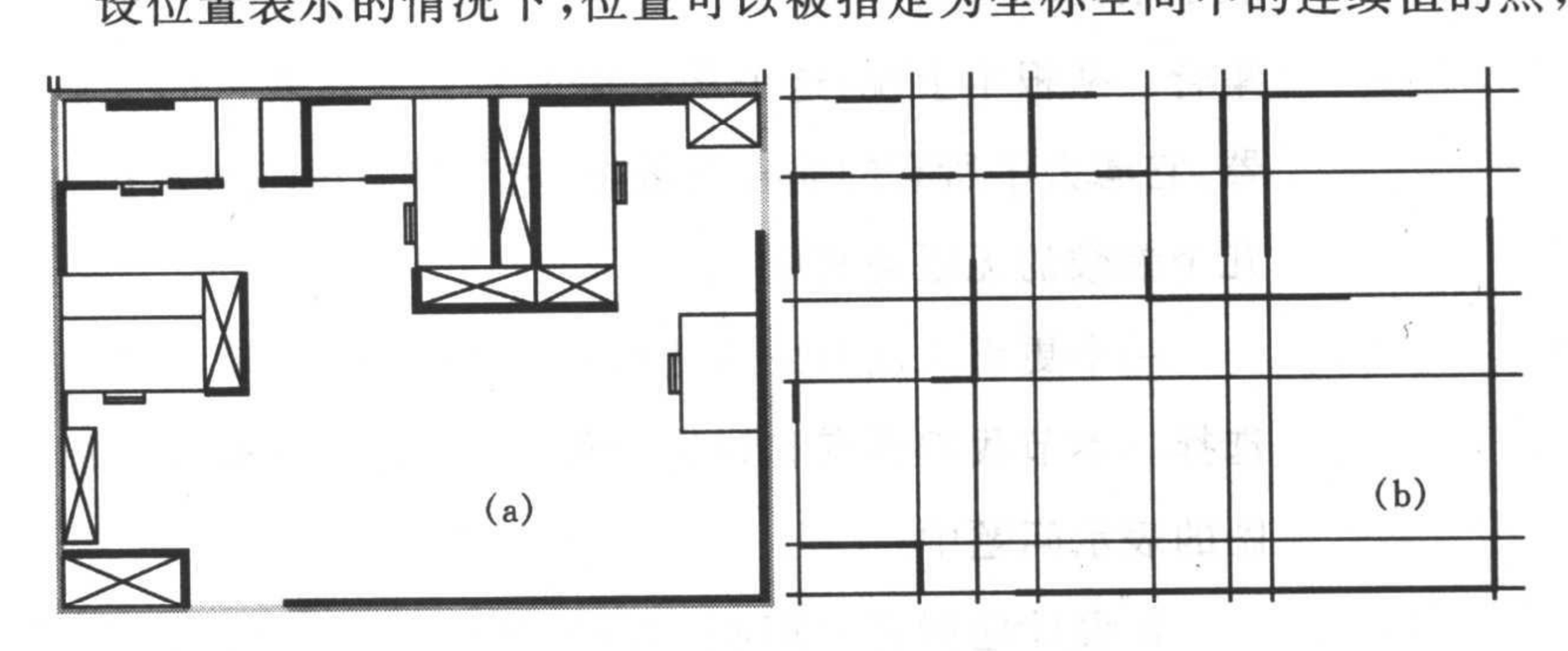


图 5.13 EPFL 连续值直线表示的实例。(a) 实际地图；(b) 用无限直线集合的表示方法

所以准确度可能极高。在多假设位置表示的情况下，连续地图使两种类型的多位置表示成为可能。

在一种情况下，机器人可能的位置可以被描述为超平面中的一个几何形状，使得我们知道机器人处在该形状内。这展示在图5.29。在该图中，机器人的位置用椭圆限定的区域予以描述。

但是，连续表示方法不是不允许以可能位置离散集合形式表示位置。例如，在参考文献[62]中，机器人位置的信任度状态，通过在机器人最佳已知位置的附近区域，对9条连续值位置进行采样而获得。该算法在一个连续空间内，获取对机器人可能位置的离散采样。

总之，连续地图表示方法的主要优点是：对于环境的配置以及该环境内机器人的位置，有高准确性和高表达性的潜能。连续表示方法的危险是地图计算上的耗费。但这个危险可以用抽象化和只获取最相关的环境特征来缓解。与闭环世界假设的使用相结合，这些技术可以使连续值地图耗费不多，甚至有时比标准的离散表示方法更便宜。

5.5.2 分解策略

在上一节，我们讨论了一个简化方法。在该方法中，连续地图的表示根据环境的2D区段，包含了近似于现实世界环境的直线无限集合。从根本上说，这个从现实世界到地图表示的变换是一个滤波器，它滤去了所有的非直线数据，进而将直线段数据扩充成只需少数几个参数的无限的直线。

一个更引人注目的简化形式是**抽象化**：环境特征的一般分解和选择。本节我们研究和讨论分解，因为它以更极端的形式应用于地图的表示问题中。

在设计地图表示期间，人们为什么要极力地分解现实环境呢？随分解和抽象化而来的缺点是地图和现实世界之间失去了保真度。既定性地根据总体结构；又定量地根据几何精度，高度抽象的地图总

不好与高保真的地图相比。

尽管这是缺点，但如果可以仔细地规划好这个抽象，使得在抛弃所有其他特征的同时获得相关的、**有用的**环境特征，分解和抽象还是有用的。这个方法的优点是地图表示可以潜在地被最小化。而且，如果分解是递阶的，例如，在一个回归抽象化的金字塔中，那么相对于地图表示的推理和规划，在计算上可以远远优越于在十分详细的环境模型中进行规划。

机会主义分解的一种标准的和无损失的形式称之为**精确单元分解**。这个由 Latombe 提出的方法，根据几何的临界性，通过选择离散单元之间的边界实现分解。

图 5.14 描述了由多边形障碍物组合成的平面工作空间的精确分解。地图表示方法将空间棋盘格化成多个自由空间区。这个表示方法可以极其精密，因为这些区域实际上是作为一个单独的节点被存储，在本例中，总共只形成 18 个节点。

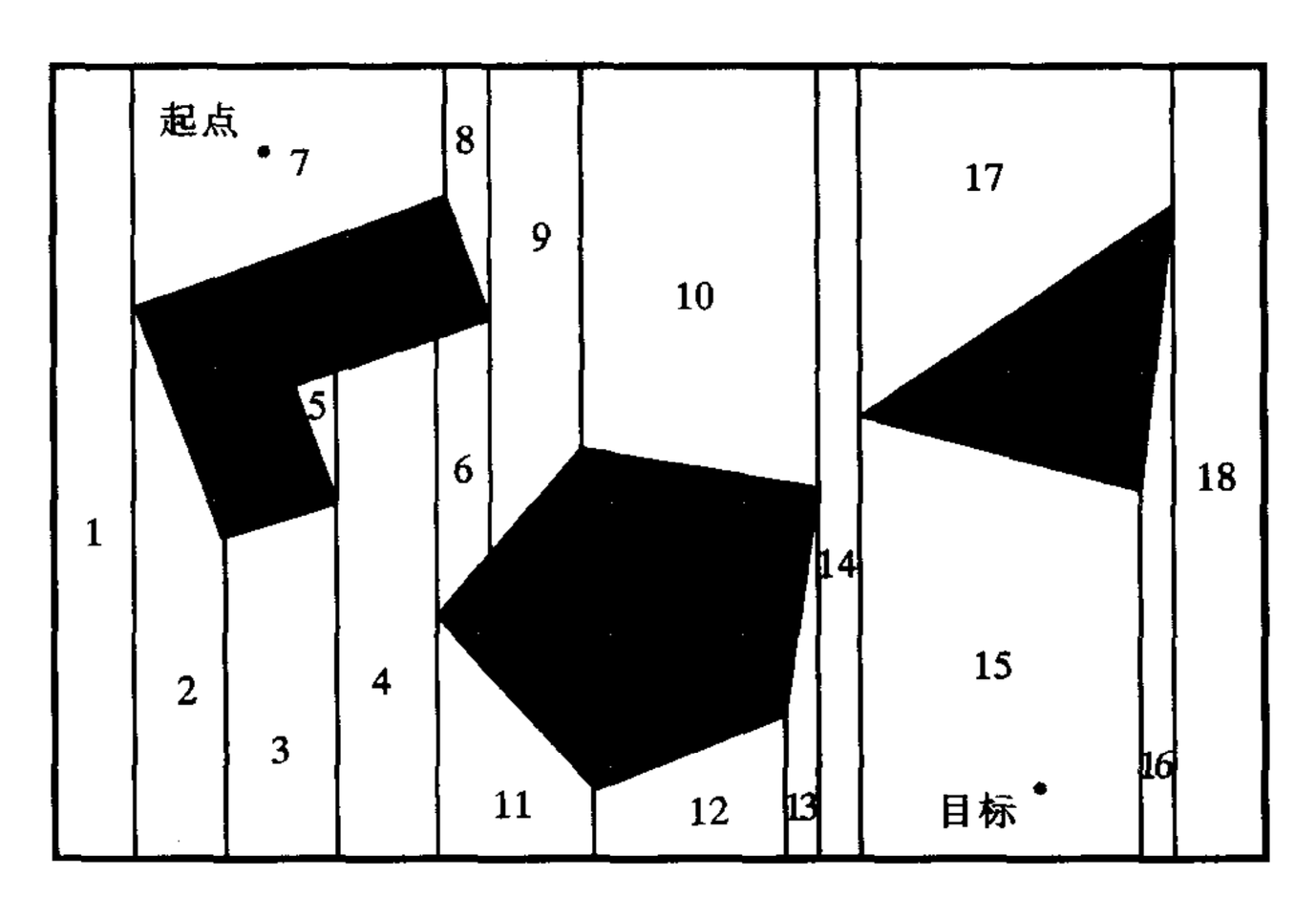


图 5.14 精确单元分解的例子

该分解背后的基本假定是在各自由控制区内，机器人的特殊位置是无关紧要的，重要的是机器人从一个自由区到相邻区的行走能力。所以，当用其他表示方法时，我们会看到最终图形保住了地图现

场的毗邻区。如果这假定确实有效，机器人并不在意单个区域内它的精确位置，那么这可以是一个有效的表示方法，而且仍然获得了环境的连接性。

这种精确的分解并不总是适当的。精确分解是特殊环境障碍物和自由空间的函数，如果信息收集贵昂，甚至不知道，则这个方法是不可行的。

另一个方法是**固定分解**。在这里，环境是棋盘格的。为了地图，将连续的现实环境变换成离散近似。该变换展示在图5.15。它描

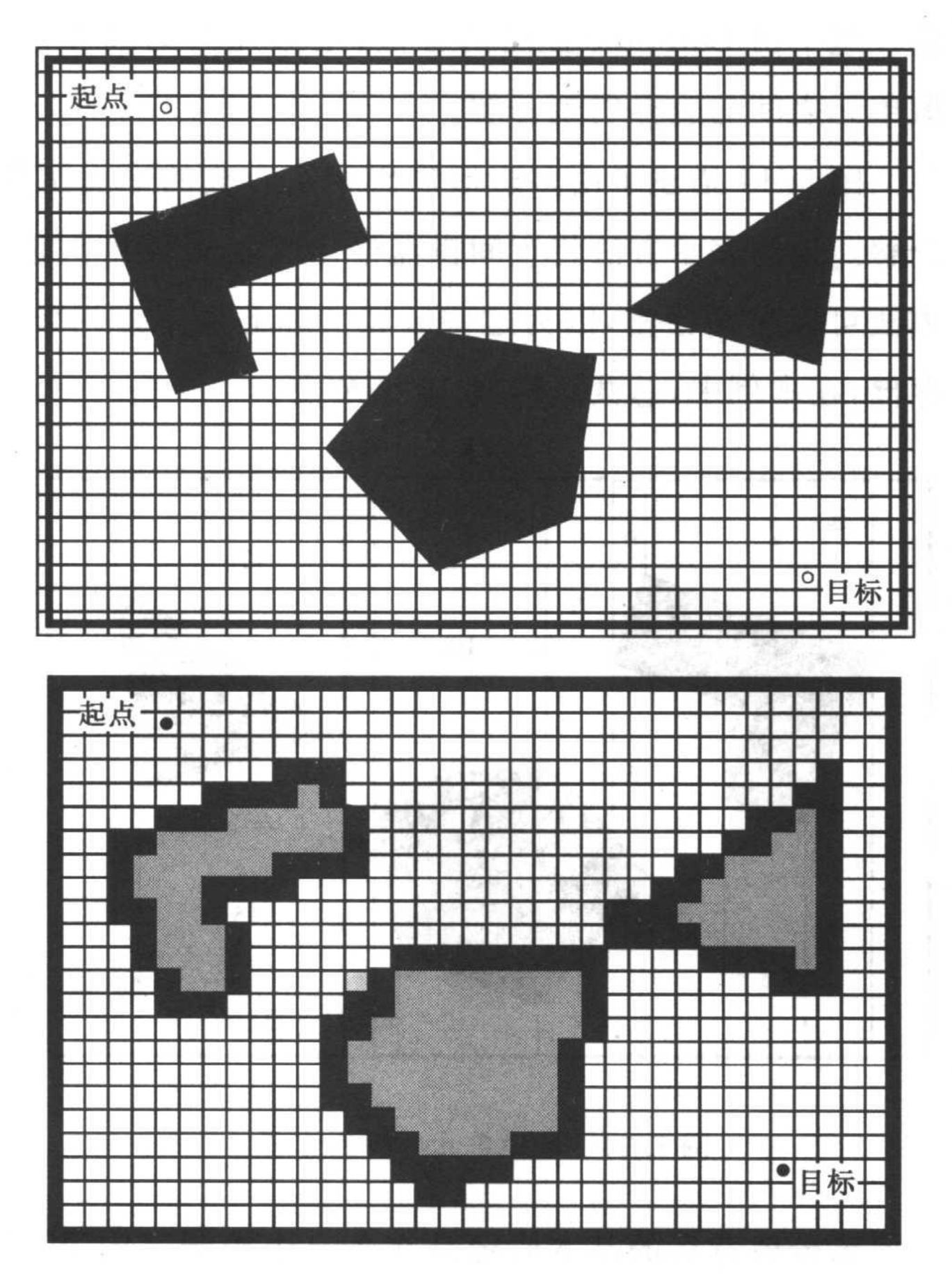


图5.15 相同空间的固定分解(窄的通道消失)

述在变换期间，填充的障碍物和自由区发生什么情况。这个方法的重要缺点来源于不精确的实质。如图 5.15 所示，在这种变换期间，窄的通道有可能丢失。从形式上来看，这意味着固定分解是合理的，但并不完全，如图 5.16 所示。还有另一个方法，即是自适应的单元分解。

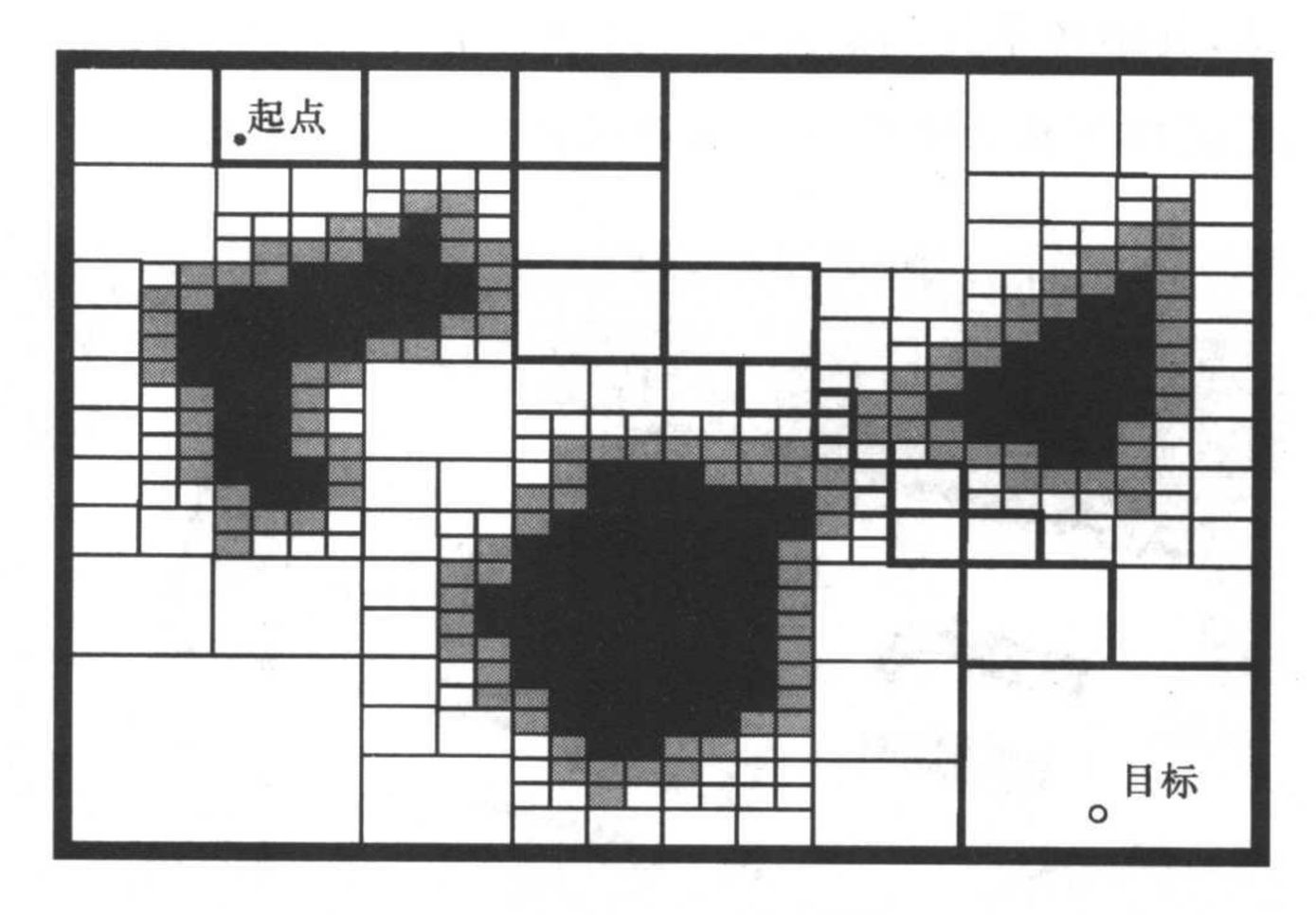


图 5.16 环境自适应(近似的可变单元)分解的例子。包围自由空间的矩形被分解成 4 个相同的矩形。如果一个矩形的内部完全位于自由空间或结构空间障碍物，那它不必再分解，否则，它递归地再分成 4 个矩形，直至得到某种预定的解。白色单元位于障碍物外面，黑色在里面，灰色是双区域部分

在移动机器人中，固定分解的概念很普遍；这大概是当前所用的单个最普通的地图表示技术。固定分解的一个非常流行的型式是所谓**占有栅格**表示方法[112]。在占有栅格中，环境被表示成离散栅格。这里，各个单元或者被填满(障碍部分)；或者是空的(自由空间部分)。当机器人装备基于测距传感器时，这特别有价值，因为各传感器的测距值与机器人的绝对位置结合在一起，可以直接被用于各单元更新填满的或空的值。

在占有栅格中，各单元可以有一个计数器。借此，0 值说明单元

还未被任何测距所“击中”，所以单元可能是自由空间。当测距点测的数目增加时，单元的值递增，超过一定的阈值，单元必定成为障碍物。当测距点**跨越**单元点测更远单元时，单元的值通常会被打折扣。用这个占有栅格方法也随时将单元的值打折扣，既可表示滞后现象，又可以表示短暂障碍物的存在可能性。图5.17描述了栅格表示方法，其中各单元的黑暗度正比于它的计数器的值。使用标准的占有栅格作图和导航的一个商业机器人是机器人Cye[163]。

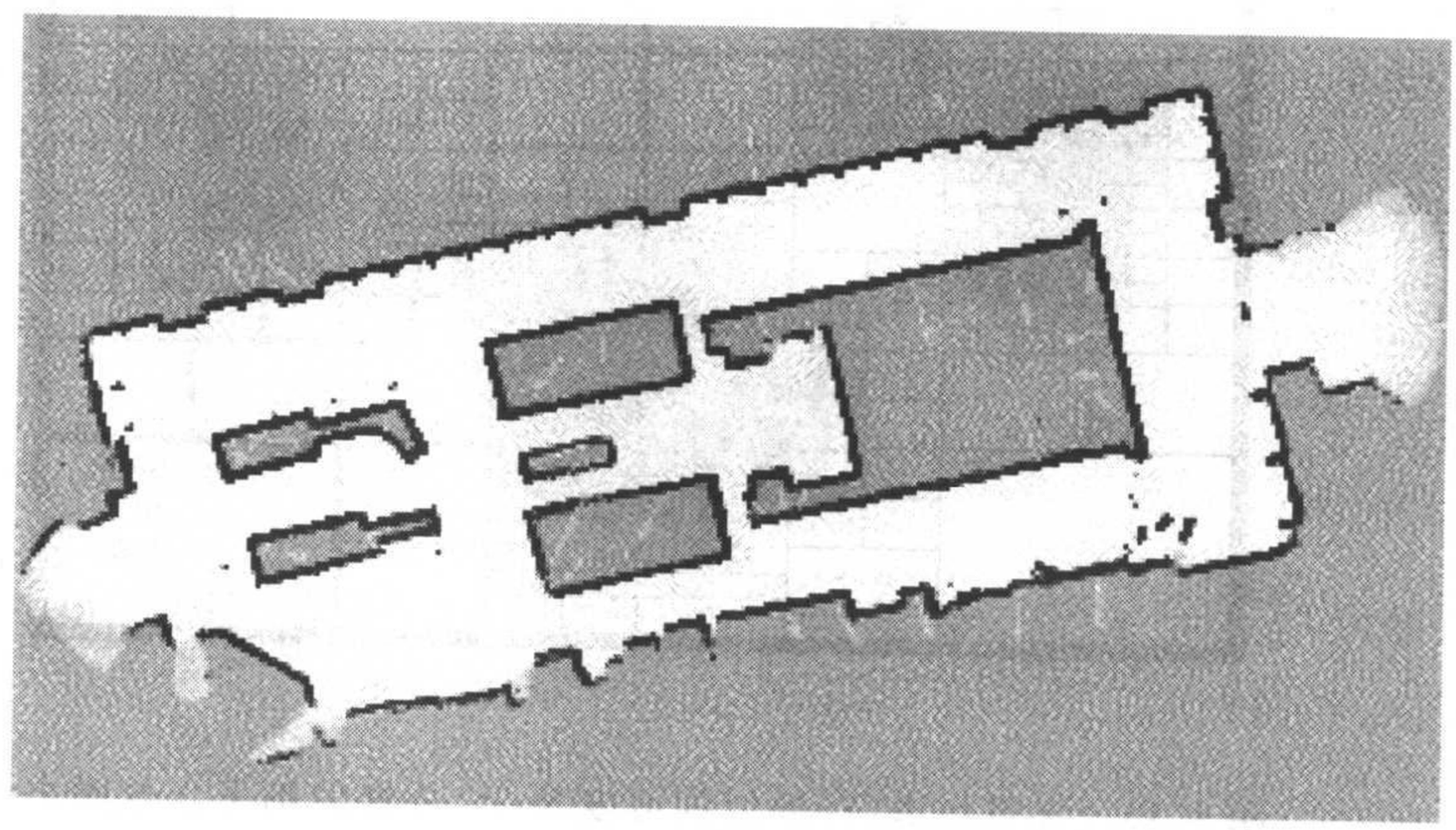


图5.17 占有栅格地图表示方法。图片由S. Thrun[145]提供

占用栅格方法有两个主要缺点。第一，在机器人存储器中地图的尺寸随着环境规模的增长而增大。如果用小的单元，尺寸迅速变得难以支持。占有栅格方法与闭环世界假定不相容，后者能使连续表示在大的、稀疏环境中具有潜在的非常小的存储需求。相反，占有栅格必须对每一个矩阵中的单元保留存储器组。而且，任何像这样的固定分解，不管环境细节如何，在环境上都要预先加上一个几何栅格。在几何特征并非环境最显著特征的情况下，这可能是不适合的。

由于这些理由，另一种方法在移动机器人中已经成为某种探索的主题，称为**拓扑**分解。拓扑方法避免了对几何环境品质的直接测量，而注重于机器人定位最相关的环境特征。

形式上，拓扑表示是指定两个对象的图形：**节点**和这些节点之间的**连接**。在拓扑表示用于移动机器人的情况下，节点用作表示环境中的区域；圆弧用作表示节点对之间的邻接。当一条圆弧连接两个节点时，则机器人可以从一个节点走到另一个节点而不必经过任何其他中间节点。

显然，如同在单元分解表示的地图邻接是现实环境的几何连接一样，邻接是拓扑方法的核心。然而，拓扑方法区别在于节点不是固定规模的，或完全是自由空间的说明，而是根据任何传感器的判断，节点提供一个区域，使得机器人能认识节点的入口和出口。

图 5.18 表示了在室内环境中，一组走廊和办公室的拓扑表示方

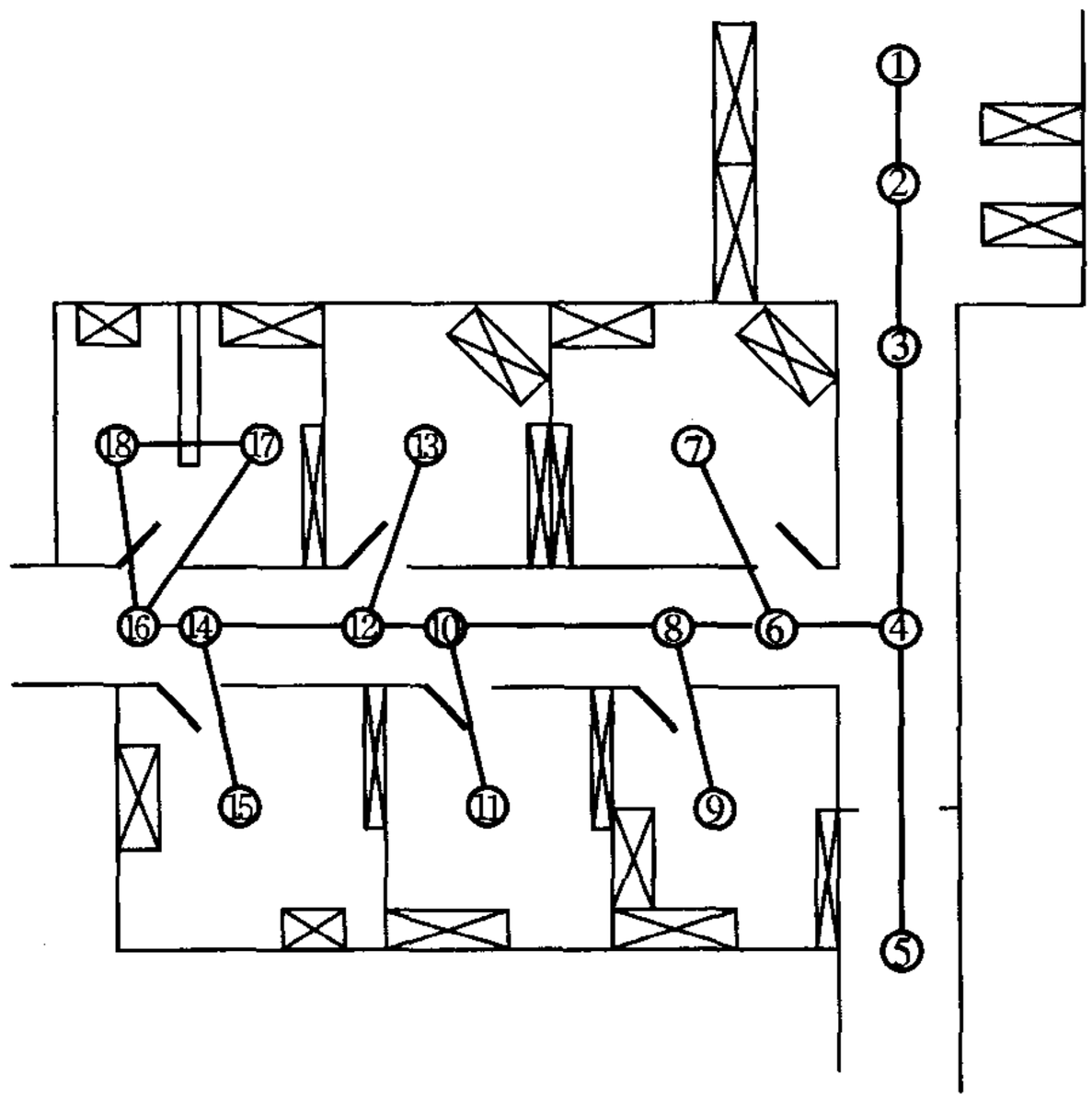


图 5.18 室内办公区的拓扑表示

法。在这种情况下，假定机器人有一个交叉检测器，例如用声纳和视觉寻找走廊之间和走廊与办公室之间的交叉口。注意到，节点占有几何空间，而圆弧在该表示方法中简单地表示为连接。

拓扑表示的另一个例子是 Simhon 和 Dudek 的工作[134]，在他们

的工作中，目标是制作能获取人类消费区最有兴趣的表象的移动机器人。在他们的表示方法中，节点是可见的引人注目的场所，而不是路径的交叉口。

为了用拓扑图鲁棒地导航，机器人必须满足两个约束：第一，它必须根据拓扑图的节点，有一个检测它当前位置的方法；第二，利用机器人的运动，它必须有一个在节点间行走的方法。节点的规模和特定的维数必须优化，使它与移动机器人硬件的传感器判断能力相匹配。对机器人特殊传感器，这个对表示进行“调节”的能力是拓扑方法的重要优点。但是，当地图表示远远偏离真正的几何图时，就失去了准确地和精密地描述机器人位置的可表达性。这里存在离散的基于单元的地图表示和拓扑表示的折衷。有趣的是，连续表示方法既有像拓扑表示那样紧凑，又有像所有直接几何表示那样精确的优势。

还有，拓扑方法的主要动机是环境可能包含重要的非几何特征——与测距无关，但有利于定位的特征。在第4章我们已描述了这种基于全图像视觉的特征。

与这些全图像特征提取器相反，通常，在环境中安放人造的空间定位路标，将环境加上特殊的可视拓扑连接性。实际上，人造的路标可加上人为的结构。用基于路标的策略，其运行的工作系统的实例也已被证明它是成功的。Latombe 基于路标的导航研究[99]，已经在现实世界实现了室内移动机器人，它使用天花板上纸做的路标作为局部可见的特征。博物馆的机器人 Chips 是另一个用人造路标解决定位问题的机器人。在该情况中，光亮的粉红正方形当作具有维数和颜色标签的路标，这种标记在博物馆的环境中是难以随意复制的[118]。图 5.19 表示了这样一种博物馆路标。

总之，距离显然不是移动机器人唯一可测和可用的环境值。特别当随着颜色视觉以及激光定位器的进展，它提供了距离信息以外的反射信息时情况更是如此。针对一个特殊移动机器人的需求选择

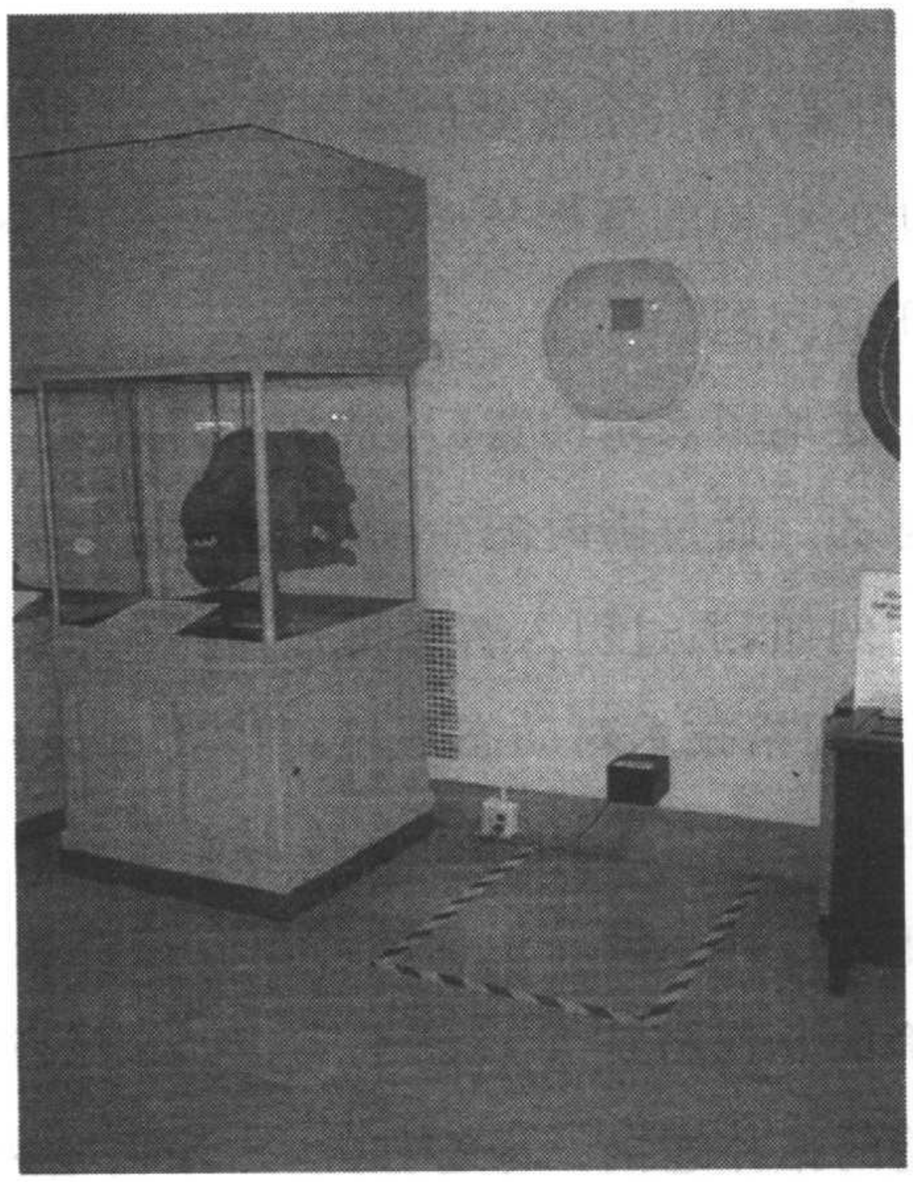


图 5.19 在自主对接期间，Chip 所用的人造路标

地图的表示方法，首先要了解移动机器人上已有的传感器；其次，要了解移动机器人的功能要求（例如，所要求的目标精度和准确度）。

5.5.3 发展水平：地图表示方法的最新挑战

以上各节阐述了有关地图表示方法选择的主要设计结果。但是，有些基本的现实世界的特征，移动机器人地图表示还没有很好地表达。它继续成为开放性研究的课题，下面我们描述几个这种带有挑战性的问题。

现实世界是动态的，当机器人进入像人类一样的空间时，它们会碰到行走的人、汽车、漫步者以及由人在他们从事活动中所安放的和移动的瞬时障碍物。当我们考虑到家庭环境时，情况确是这样。在这环境里，终会有一天需要争用家庭机器人。

一般而言，上述的地图表示，没有明显的办法识别和区分永久性障碍物（如墙、走廊等）和临时障碍物之间的差别。移动机器人传感

器的当前发展水平，部分地归咎于这个缺点。虽然视觉研究正在迅速进步，但是，从**一个移动的参考框架**，还没有可用的区分运动的动物和静止的结构的鲁棒传感器。而且，估计瞬态对象的运动向量仍是一个研究问题。

在上述地图表示的背后，常常假定所有在地图上的物体实际上都是静态的。通过随时减少所画物体的办法，上述方法可以获得部分成功。例如，通过引入暂时折扣，有效地将瞬时障碍物当作噪声来处理，占用栅格技术对动态环境可以更加鲁棒。地图制作更具挑战性的过程是它对环境的动力学特别地脆弱，大多数的作图技术一般要求在作图过程中没有运动的对象。拓扑表示方法没有这种限制，因为精确的几何特征并不重要，主要承受的约束是瞬时对象必须不改变环境的拓扑连接。尽管如此，占用栅格表示方法和拓扑方法二者都不主动地识别和表示瞬时对象，因为瞬时对象既有别于传感器的误差，也不同于永久性的地图特征。

视觉感知提供有关环境中对象的瞬态和运动细节的更鲁棒和更丰富的信息内容。所以，移动机器人专家及时地提出了利用那些信息的表示方法。典型的例子包括人群造成的堵塞。博物馆导游机器人通常承受大量的堵塞。如果机器人的感知装置沿着机器人身体安装，那么当一群参观者完全包围机器人时，机器人实际上成了瞎子。因为它的地图只包含环境特征，这时，由于人墙的影响，机器人的传感器完全感知不到环境特征。在最佳的情况下，机器人应该认知它的堵塞，且不利用这些无效的传感器读数进行定位。在最坏的情况下，机器人会用完全堵塞的数据来定位，并会不正确地更新它的位置。可判断机器人局部条件（例如被人包围）的视觉传感器，可以帮助消除这个错误模式。

移动机器人定位的第二个开放的挑战性问题涉及开放空间的穿越。现有的定位技术一般依赖于局部的测量，诸如距离。因此，要求环境比较密集地充实传感器可检测和可测量的物体。开阔的空间，

如停车场、草地和诸如在会议中心见到的室内门廊，由于对象比较稀疏，就会对这种系统造成困难。的确，当有人居住时困难加剧，因为任何被映入的物体几乎肯定地要被人挡住视野。

更新近的技术对克服这些缺点再次提供了某些希望。视觉和新近的激光测距装置二者提供了距离远至 100 m 或更多的室外特性指标。当然，GPS 性能更佳。对利用距离特征进行定位的机器人，也许要求长距离的感知。

这种倾向挑明了最基本的拓扑地图表示方法的隐含假定。通常，拓扑表示方法就空间能力作了假定：节点包含了节点内它们本身的物体和特征。因此，地图制作过程包含了制作节点。这些节点以它们自含的方式，依靠节点内所包含的物体，是可以识别的。所以，在室内环境，各室可以是单独的一个节点，这是合理的，因为各室会有一个对该室唯一的布局和一组所属物。

然而，我们考虑一个开阔的室外公园环境。哪里应该是单独节点的终端？下一个节点从哪里开始？回答是不清楚的，因为远离当前节点或位置的物体，可以为定位过程获得信息。例如，地平线上的山峰、山谷中的河流和太阳的轨迹全都是非局部的特征，它们具有推断当前位置的相关能力。这违反了空间局部性的假定，代之以可视性准则：节点或单元需要有表示物体的机制，该物体从那单元出发是可测的和可见的。再次，随着传感器的改善，并在这种情况下随着室外运动机制的改善，将有更大的紧迫性来解决在开阔环境中（使用或不使用 GPS 类型的全局定位传感器）与定位相关的问题。

我们以最后一个挑战性问题结束这一节。这问题表述了机器人学基础性学术研究的一个问题：传感器融合。使用现货供应的机器人传感器，包括热、距离、音频和基于光的反射性、颜色、纹理、摩擦等各种不同类型的测量都是可能的。传感器融合是与地图表示方法密切相关的研究课题。正如为了机器人执行定位和推理，地图必须足够详细地具体表达环境一样，传感器融合也要求有环境的表示，这表

示要有足够的通用性和表现力，使各种传感器类型能具有它们合适的相关数据，增强了最后合成的感知，令人满意地超越任何单个传感器的读数。

大概至今为止，传感器融合唯一通用的实现方法是神经网络分类器。利用这个技术，任何数目和任何类型传感器的值都可以被联合组合成一个网络，它使用无论什么必需的方法来优化它的分类准确度。对必须使用人类可读的内部地图表示方法的移动机器人，还没有诞生出合适通用的传感器融合方案。当传感器融合问题得到解决时，我们有理由期待，完全不同类型的大量数目的传感器的集成可以为机器人实现现实世界的导航，甚至在开阔和动态环境中（如占满人的公共广场）的导航，并容易地形成足够的判别能力。

5.6 基于概率地图的定位

5.6.1 引言

如前所述，多假设位置表示方法是有优点的，因为机器人可以明确地跟踪环境中有关它可能位置的自己信任度。在理想状态下，机器人的**信任度状态**会随着时间而改变，与电机的输出和感知的输入维持一致。前面指出，多假设表示的一个几何方法是通过指定环境表示[98]的多边形，涉及辨识机器人的可能位置。这个方法对不同机器人的可能位置之间的相对选择机会，不提供任何说明。

概率技术与此不同，因为它明确地给出了可能的机器人位置的辨识概率。因此，这些方法已成为当前研究的焦点。下面，我们提出两类概率定位的方法。第一类，**马尔可夫定位**，对所有可能的机器人位置使用明确指定的概率分布；第二个方法，**卡尔曼滤波定位**，使用机器人位置的高斯概率密度的表示和定位匹配的扫描。与马尔可夫

定位不同，卡尔曼滤波定位在机器人配置空间中不独立地考虑各个可能的姿态。有趣的是，如果假定机器人位置的不确定性具有高斯形式[3,pp.43-44]，则卡尔曼滤波定位过程可从马尔可夫定位的公理导出。

在详细讨论各个方法之前，我们提出一般的机器人定位问题和求解策略。让我们考虑在一个已知环境中的移动机器人。当它开始移动时，例如，从一个精确知道的位置开始，利用里程表它可以保持对它运动的跟踪。由于里程表的不确定性，移动几步之后，机器人对它的位置会变得很不确定(见 5.2.4 节)。为了使位置不确定性不致无限增长，机器人必须相对于它的环境地图将它自己定位。为了进行定位，机器人会使用它内装的传感器(超声、距离传感器、视觉)对环境进行观测。由里程表提供的信息加上这种外感受的观测所提供的信息，可以被组合起来，使机器人尽可能好地相对于地图定位。根据本体感受传感器的值和外感受传感器的值，其更新过程常常在逻辑上是分离的，这就产生了一般机器人位置更新的二步过程。

动作更新表示将某动作模型 Act 应用于移动机器人的本体感受编码器的测量 o_t 和先验的信任度状态 s_{t-1}，得到表示机器人关于当前位置信任度的新信任度状态。注意到，整个这一章我们随时假定，机器人本体感受编码器的测量都被用作其动作最佳的可能测量。例如，如果差动驱动的机器人有电机(不连接到轮子的编码器)并采用开环控制，那么，必须对轮子的旋转实施机器人的高度不确定的估计，而不是用编码器的测量。我们忽略这些情况，故有以下简单方程：

$$s'_t = Act(o_t, s_{t-1}) \tag{5.16}$$

感知更新表示将某感知模型 See 应用于移动机器人的外感受传感器 i_t 和已更新的信任度状态 s'_t，得到表示机器人当前位置的精确的信任度状态：

$$s_t = See(i_t, s'_t) \tag{5.17}$$

感知模型 *See* 和有些时候的动作模型 *Act* 二者都是地图和机器人物理配置(如传感器和它们的位置,运动学等)的抽象函数。

一般动作更新过程给机器人有关位置信任度提供不确定性:编码器有误差,从而运动有相当程度的不确定性。相反,感知更新通常改善信任度状态。传感器的测量,在与机器人环境模型比较时,有助于提供关于机器人可能位置的线索。

在马尔可夫定位的情况下,对地图中每一个可能的机器人姿态,经常将机器人信任度状态表示成单独的概率分配。在这种情况下,动作更新和感知更新必须更新每一个单元的概率。卡尔曼滤波器定位用单独的、充分定义的高斯概率密度函数表示机器人的信任度状态。因此,正好保留了有关相对于地图位置的机器人信任度的 μ 和 σ 的参数化。全部的要求就是更新高斯分布的参数。在信任度状态表示方法中,这个根本性的差别,如文献[73]所提出的,导出了这两个方法以下的优点和缺点:

- 马尔可夫定位考虑了从任何未知位置开始的定位,因而可以恢复不明确的状况,因为机器人能够跟踪多个完全不同的可能位置。然而,要在任何时刻,在整个状态空间内更新所有位置的概率,就需要空间的离散表示,诸如几何栅格或拓扑图(见5.5.2节)。所以,其所要求的存储器和计算功能可能限制了精度和地图尺寸。

- 卡尔曼滤波器定位从起始的已知位置开始跟踪机器人,具有固有的精确和有效性质。特别是卡尔曼滤波器定位可以被用在连续环境表示中。然而,如果机器人的不确定性变得太大(例如,由于机器人与物体碰撞),并因此造成非真正的单模,卡尔曼滤波器就不能捕获多个可能的机器人位置,从而可能有无法挽回的损失。

在最近的研究项目中,或者在马尔可夫方法[49]内只更新感兴趣的状态空间;或者用卡尔曼滤波器跟踪多假设[35];或者将两个方法结合起来建立混合的定位系统[73,147],从而实现或提出了改进。在下面两节,我们将详细描述各个方法。

5.6.2 马尔可夫定位

马尔可夫定位利用任意概率密度函数表示机器人的位置[50,88,116,119]，跟踪机器人的信任度状态。实际上，所有已知的马尔可夫定位系统首先将机器人的配置空间棋盘栅格化成地图中有限的、离散数目的可能机器人姿态，以此来实现这个通用的信任度表示。在实际应用中，可能姿态的数目可以从几百个到上百万个位置的范围内变化。

给定这种机器人位置的一般概念需要强有力的机制。当新的信息（例如编码器和传感器的值）以任意的概率密度并入先验的信任度状态时，可以计算所造成的信任度状态。求解方法来源于概率论。所以，在下节我们描述了应用于此问题的概率论基础，即著名的贝也斯方程。接下来的两节给出实例研究，一个机器人使用简单的、特征驱动的环境拓扑表示[88,116,119]；另一个机器人利用基于几何栅格地图[49,50]。

5.6.2.1 引言：概率论应用于机器人定位

给定机器人位置的离散表示，为了表达信任度状态，我们希望给各个可能的机器人位置分配一个机器人确实在该位置的概率。按概率论，我们用 $p(A)$ 表示 A 是真的概率。这也叫 A 的先验概率，因为它测量了 A 是真的概率，而与我们拥有的任何附加的知识无关。例如，我们可以用 $p(r_t=l)$ 表示在时刻 t 机器人在位置 l 的先验概率。

实际上，我们希望，给定机器人已收集的编码器和传感器证据，计算各单个机器人位置的概率。在概率论中，我们用 $p(A|B)$ 表示我们知道 B，A 的**条件**概率。例如，我们用 $p(r_t=l|i_t)$ 表示给定机器人的传感器输入 i_t，机器人在位置 l 的概率。

问题是：像 $p(r_t=l|i_t)$ 这样的项，如何能被简化成它的组成部分，使得我们能计算它？其答案在乘积律，它表明：

$$p(A \wedge B)=p(A|B)p(B) \tag{5.18}$$

方程(5.18)是直觉的,因为 A 和 B 二者为真,与 B 为真和其他有条件的为真相关。但你应该自己能确信,另一个方程同样是正确的:

$$p(A \wedge B) = p(B|A)p(A) \tag{5.19}$$

方程(5.18)和(5.19)合起来使用,我们可以导出计算 $p(A|B)$ 的贝也斯方程:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{5.20}$$

我们用贝也斯规则计算机器人新的信任度状态,它是它的传感器输入和以前信任度的函数。为了理想地计算,我们必须回忆马尔可夫定位的基本目标:表达可能的机器人位置 L 的离散集合。机器人的信任度状态必须给 L 中各个位置 l 分配一个概率 $p(r_t = l)$。

在方程(5.17)中描述的 *See* 函数表达了从一个信任度状态和传感器输入到精确的信任度状态的映射。为此,我们必须更新与 L 中各位置 l 相关的概率。通过直接把贝也斯方程用于每一个这样的 l,我们就可以实现更新。在标记方面,为了简化,我们将不用时间指数 t,进而用 $p(l)$ 代表 $p(r=l)$:

$$p(l|i) = \frac{p(i|l)p(l)}{p(i)} \tag{5.21}$$

$p(i|l)$ 的值对方程(5.21)是关键,这个在各机器人位置上的传感器输入的概率必须用某种模型进行计算。一个显而易见的策略是参考机器人的地图,给定有关机器人传感器几何特征和被映射的环境,利用各可能的地图位置辨识特定传感器读数的概率。$p(l)$ 的值在该情况下容易重新获得。它简单地是感知更新过程之前,与信任度状态有关的概率 $p(r=l)$。最后,注意到,分母 $p(i)$ 与 l 无关;即当我们把方程(5.21)应用于 L 中所有位置 l 时,分母从不改变。因为,它是有效的常数。实际上,该分母常常减少。在感知更新步骤的终点,信任度状态的所有概率被重新归一化使和为 1.0。

现在考虑方程(5.16)的 *Act* 函数,*Act* 将前一个信任度状态和

编码器测量(即机器人动作)映射成新的信任度状态。为了计算在新信任度状态下位置 l 的概率,我们必须根据在前一信任度状态中所表达的潜在位置,对机器人可到达 l 所有可能的路程求积分。这是错综复杂的,但从根本上说也是重要的。从具有相同的编码器测量 o 的多个源位置出发可以到达同一位置 l,因为编码器测量是不确定的。在这更新方程中,需要时间指数:

$$p(l_t \mid o_t) = \int p(l_t \mid l'_{t-1}, o_t)\, p(l'_{t-1})\, \mathrm{d}l'_{t-1} \tag{5.22}$$

因此,给定编码器测量 o,对特定位置 l,总的概率是由前一个信任度状态中每个位置 l' 各自的贡献而建立起来的。

方程(5.21)和(5.22)形成了马尔可夫定位的基础,而且它们体现了**马尔可夫假设**。形式上,这意味它们的输出只是机器人前一状态和它的最近动作(里程表)与感知的函数。一般而言,在非马尔可夫情况下,其系统的状态依赖于它经历的所有状态。毕竟在时刻 t,机器人传感器的值不仅仅只依赖于在时刻 t 它的位置。它们在某种程度上依赖于整个时间机器人的轨迹;的确,依赖于机器人整个经历。例如,机器人最近可能经历了一次严重的碰撞,使得它产生一个有偏的传感器行为。相同地,在时刻 t,机器人的位置真的不只依赖于它时刻 $t-1$ 的位置和它的里程表测量。由于它的运动经历,一个轮子可能比另一个轮子磨损得更厉害些,造成整个时间左转偏置,从而影响它当前位置。

当然,马尔可夫假设不是一个有效的假设。然而马尔可夫假设大大简化了跟踪、推理和规划,它在移动机器人学中继续成为非常普遍的近似假设。

5.6.2.2 实例研究 1:用拓扑图的马尔可夫定位

当机器人的环境表示方法已经提供合适的分解时,可以直截了当地应用马尔可夫定位。在环境表示是纯粹的拓扑时,就是这种情况。

考虑一个比赛,在比赛中,各机器人接受环境的拓扑描述,描述

将只包括走廊和房间而不指出几何距离。另外,该所提供的**地图**是不理想的,包含了几个错误的弧(例如关闭的门)。这正是1994年美国人工智能协会(AAAI)全国机器人比赛的情况。在那次比赛中,各机器人的任务是利用所提供的地图和它自己的传感器,从所选择的开始位置导航到目标室。

这次比赛的优胜者Dervish使用了概率的马尔可夫定位,且在整个拓扑环境表示中利用多假设的信任度状态。我们现在描述Dervish,把它作为用离散拓扑表示和概率定位算法的机器人实例。

如图5.20所示,Dervish包含了为1994年AAAI全国机器人比赛专门设计的声纳装置。在这次比赛中,环境由直线式室内办公室组成,内有实际家具当作障碍物。传统的传感器环绕机器人按径向安装成一个环。具有这种传感器配置的机器人既会因低于环的短小物体而绊倒,又会因高于环的高大物体(诸如壁架、架子和装饰板)而被碰倒。

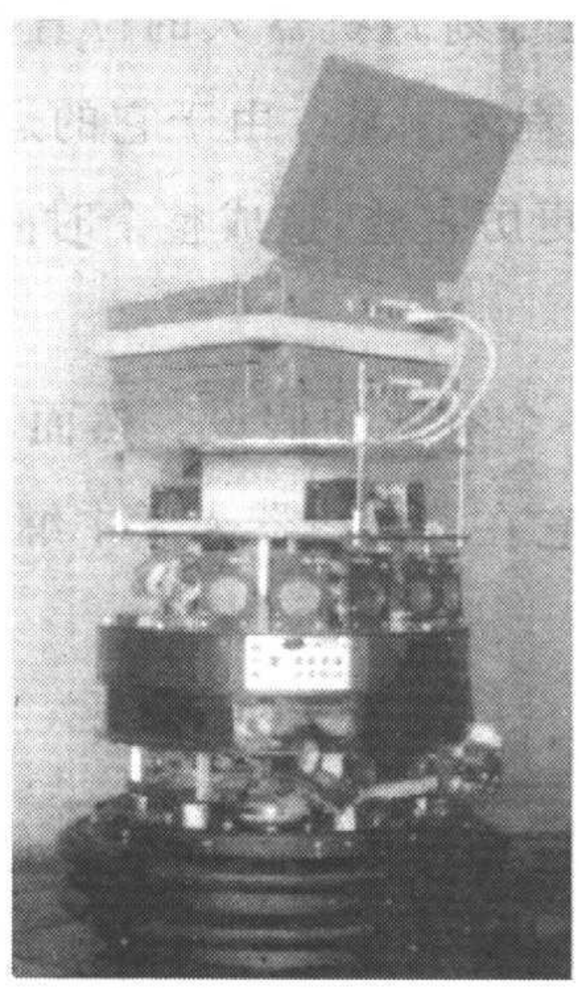


图5.20 Dervish探索它的环境

Dervish 解决此困难的办法是：将一对传感器对角地朝上安放，以检测壁架和其他悬挂物。另外，这一对的对角传感器也被证明能够巧妙地检测装饰板，使机器人避免在高的装饰板下徘徊。其他的传感器聚合成传感器集合，使得集合中单个变送器处在稍微不同的角度，让反射性最小。最后，接近基部安装两个传感器，以检测低的物体，如地板上的纸杯。

我们已经注意到，比赛组织者所提供的表示方法是纯拓扑的，它标注办公环境中走廊和房间的连接性。因此，这适合于设计 Dervish 的感知系统以检测对感知事件所进行的匹配：即探测和穿过走廊和办公室之间的接口。

这个**抽象**的感知系统在整个时间内是由观察声纳对 Dervish 左边和右边轨迹的探测而完成的。有趣的是，该感知系统会单独使用时间，没有编码器值触发感知事件的概念。例如，当机器人在走廊宽度中，连续地 1 秒多时间里检测到 7～17 cm 缺口，就触发**门关**传感器事件。如果，在 1 秒多时间里声纳探测到突变区超过17 cm，**门开**传感器事件触发。为了减少与 Dervish 声纳有关的固有的反射传感器的噪声（见 4.1.6 节），机器人跟踪它相对于走廊中心线的角度，而且当它相对于走廊的角度超过 7°时，完全抑制传感器事件。有趣的是这会产生一个常常丢失特征的保守的感知系统，特别当 Dervish 必须导航的走廊挤满障碍物时。再者，感知系统的保守性质，特别是它引发虚假导航的趋向，会对定位问题指出了一个概率的解决方案，以使我们可以考虑感知输入的完整轨迹。

Dervish 的环境表示是一个离散的拓扑地图，在抽象化和信息方面与比赛组织者所提供的地图相同。图 5.21 刻画了一个典型的办公室环境的几何表示，旁边并列着同一办公室环境的拓扑图。回忆一下，对一个拓扑表示，其关键的决策包括节点的分配和节点间连接性（见 5.5.2 节）。如图 5.21 左边所示，Dervish 用了一个拓扑，在拓扑中，节点边界主要由门口（以及走廊和大厅）进行标记。右边所示

的拓扑图说明了在所示例子中所获得的信息。

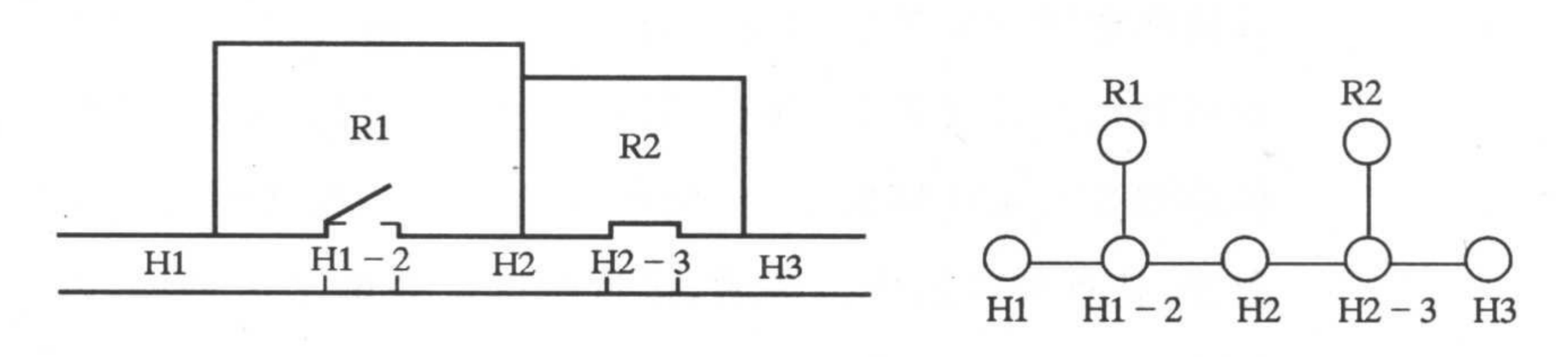


图 5.21 几何的办公室环境(左)和它的拓扑模拟(右)

注意,在这个特殊的拓扑模型中弧是零长的,而节点具有空间扩张,合在一起覆盖了整个空间。这个特殊拓扑表示方法特别适合于Dervish,它给出了经过走廊进入指定房间的导航任务和它辨识走廊墙壁断续性的感知能力。

为了表示指定的信任度状态,Dervish将各拓扑节点 n 与机器人处在 n 边界内的一个物理位置的概率 $p(r_t=n)$ 关联起来。下面我们会清楚,Dervish所用的概率更新是近似的。所以,技术上人们应该把最终值称做**似然值**而不是概率。

表 5.1 Dervish 的确定性矩阵

	墙	门关	门开	走廊开放	大厅
检测不到东西	0.70	0.40	0.05	0.001	0.30
检测到门关	0.30	0.60	0	0	0.05
检测到门开	0	0	0.90	0.10	0.15
检测到走廊开放	0	0	0.001	0.90	0.50

Dervish感知更新过程精确地按方程(5.21)运行。感知事件异步地产生,在各个时刻,特征提取器能够根据最近的超声值辨认大尺寸的特征(如门口、交叉口)。各个感知事件由“感知对”组成(机器人一侧一个特征或两侧两个特征)。

给定一个特定的感知对 i,利用方程(5.23),方程(5.21)能够使各可能位置 n 的似然值更新。

$$p(n|i)=p(i|n)p(n) \tag{5.23}$$

$p(n)$的值，从 Dervish 当前信任度状态已经知道，所以问题在于计算 $p(i|n)$。Dervish 的主要简化是基于如下现实的：因为特征提取系统只提取 4 个节点类型，又因为一个节点包含（在一个边上）5 个总特征之一，所以节点类型和所提取特征的每一个可能组合可以表示成 4×5 的表格。

Dervish 的**确定性矩阵**（见表 5.1）正是这个查询表。Dervish 提出简化的假设，即特征检测器的特性指标（即正确的概率）仅仅是所提取的特征和节点中实际特征的函数。有了这个假设，我们可以从各感知和节点类型的可能配对，给出**确定性矩阵**置信度的估计。对机器人可能碰到的 5 个环境特征的各个特征（墙、门关、门开、走廊和大厅开放），该矩阵对传感器系统可以发出的 3 个单侧感知的各事件分配一个似然值。此外，该矩阵也给传感器系统完全发觉不到的感知事件（检测不到东西）分配似然值。

例如，利用表 5.1 的指定值，如果 Dervish 与一个开放走廊相邻，错误地辨认它为门开的似然值是 0.10。这意味，对**开放走廊**类型的任何节点 n，和对传感器的值 i=门开，$p(i|n)=0.10$。连同一个特殊的拓扑图，确定性矩阵能够在感知更新过程中直截了当地计算 $p(i|n)$。

对 Dervish 的特殊传感器组和它意向导航的任何特定环境，人们产生了特定确定性矩阵。连同全局的、在现实环境中任何给定的门是开还是关的概率测量，该矩阵不严格地表示了感知的置信度。

回忆一下，Dervish 没有编码器，感知事件异步地由特征提取过程触发。所以，Dervish 没有像方程（5.22）那样有动作更新步骤。当机器人确实检测到一个感知事件时，就需要执行多个知觉的更新步骤。在给定 Dervish 前一个信任度状态的情况下，需要更新每一个可能的机器人位置的似然值。这是因为从它的以前感知事件（即，假阴性误差）起，存在机器人走过**多个**拓扑节点的机会。形式上，Dervish 感知更新方程实际是动作更新一般形式和感知更新的组合。给

定感知事件 i,位置 n 的似然值按方程(5.22)计算:

$$p(n_t \mid i_t) = \int p(n_t \mid n'_{t-i}, i_t) p(n'_{t-i}) \mathrm{d}n'_{t-i} \tag{5.24}$$

如同用 Dervish 的前一信任度状态所表示的那样,$p(n'_{t-i})$的值标记 Dervish 在位置 n'的似然值。时间下标 $t-i$ 用在 $t-1$ 场合,因为对各个可能位置 n',从 n'到 n 的离散拓扑距离,可以按指定的拓扑地图而变化。$p(n_i \mid n'_{t-i}, i_t)$用位置 n 产生感知事件 i 的概率,乘上 n'和 n 之间的所有节点产生感知事件失败的概率进行计算:

$$p(n_t \mid n'_{t-i}, i_t) = p(i_t, n_t) \cdot p(\phi, n_{t-1}) \cdot p(\phi, n_{t-2}) \cdot \cdots \cdot p(\phi, n_{t-i+1}) \tag{5.25}$$

对于例子(图 5.22),假定机器人在信任度状态中只有 2 个非零节点{1-2,2-3},各具有与各可能位置相关的似然值:$p(1-2)=1.0$和 $p(2-3)=0.2$。为了简单,假定机器人确定性地面向东。注意到,节点 1-2 和 2-3 似然值之和不等于 1.0。这些值不是正式的概率,在 Dervish 中,由于完全避免了归一化,使计算工作量最少。现在我们假定,产生了一个感知事件:机器人同时检测到它的左边是开放的走廊,右边是打开的门。

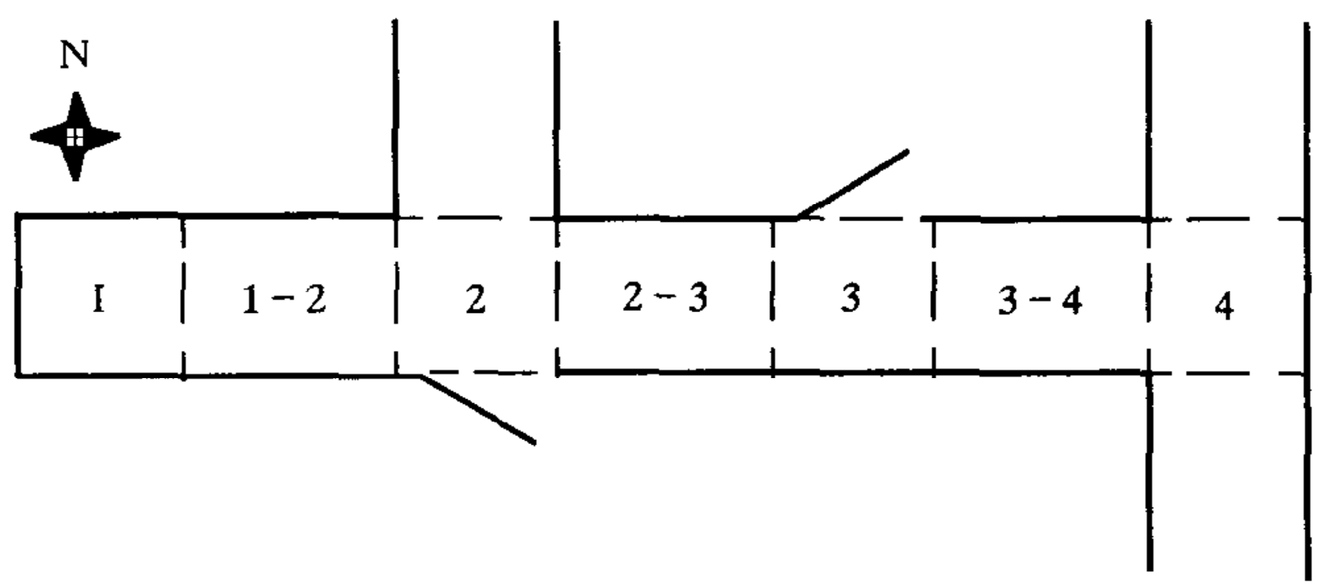


图 5.22 实际室内拓扑环境

状态 2-3 将可能进行到状态 3,3-4,和 4,但状态 3 和 3-4 可能被消掉,因为当只有墙时,检测到打开的门的似然值是零。到达状态 4 的似然值是状态 2-3 的初始似然值 0.2,在节点 3 检测不到任何东西的似然值(a)和在节点 4 左边检测到走廊、右边检测到门的似

然值(b)的乘积。注意,我们假定在节点 3－4 检测不到任何东西的似然值为 1.0(简化近似)。

(a) 只发生在如果 Dervish 在节点 3 检测不到它左边的门(关闭或开),[0.6·0.4+(1−0.6)·0.05],和在它的右边正确地检测不到物体,0.7;

(b) 发生在如果 Dervish 正确地在节点 4 辨识开放的走廊,0.9,以及把开的门错误当作走廊,0.10。

最后的方程,0.2·[0.6·0.4+0.4·0.05]·0.7·[0.9·0.1],得到状态 4 的似然值 0.003。这是从先验信任度状态节点 2－3推出来的 $p(4)$的部分结果。

转到 Dervish 先验信任度状态中其他节点,1－2 将可能进行到 2,2－3,3,3－4 和 4。再者,状态 2－3,3,和 3－4 可能全被消除,因为当墙出现时,检测到打开的门的似然值为零。状态 2 的似然值是以下三部分的乘积:状态 1－2 先验似然值 1.0,检测到右边的门为打开的门的似然值[0.6·0+0.4·0.9],和正确检测到向左的开放走廊的似然值 0.9。因此处在状态 2 的似然值为 1.0·0.4·0.9·0.9=0.3。此外,1－2 以 $4.3\cdot10^{-6}$的确定性因子进行到状态 4,将此因子加到上述的确定性因子,状态 4 的总的因子为 0.003 28。所以 Dervish 跟踪新的信任度状态为{2,4},给位置 2 分配很高的似然值;给位置 4 分配一个低的似然值。

经验上,Dervish 的地图表示和定位系统已证明足以导航四个室内办公室环境:明确地为 1994 年美国全国人工智能会议建立的人工办公环境;斯坦福大学心理学、历史、和计算机科学建立的办公环境。当不给 Dervish 提供相邻节点之间的距离概念时,所有这些实验都是在 Dervish 的拓扑地图中运行。这是概率定位强有力的展示,尽管非常缺少动作和编码器的信息,但机器人能够成功地导航几个现实世界的办公大楼。

对于 Dervish 的定位系统,有一个仍未解决的问题。Dervish 不

仅仅是定位器，它还是一个导航器。如对所有多假设系统一样，人们必定会问这样的问题：在它的表示中，给定机器人多个可能的位置，机器人如何决策？应怎样移动？Dervish 所用的技术，是移动机器人学领域普通的技术：假定机器人的实际位置是在信任度状态中最可能的节点，据此来规划机器人的动作。一般而言，最可能位置是机器人实际环境位置的良好度量。然而，当最高和次高的最可能位置具有相同值时，该技术就有缺点。在 Dervish 的例子下，尽管它总是与最高的似然值位置配合在一起，保留在一个关键的结合点，但机器人的目标是进入目标房间并停留在那里。所以，从它的目标观点出发，重要的是，只有当机器人有处在正确的最终位置的强置信度时，Dervish 才完成导航。在这种特殊情况下，如果最可能位置和次似然值位置之间的差异低于预定的阈值，Dervish 的执行模块拒绝进入房间。在这种情况下，Dervish 会主动规划一条路径，使机器人进一步移动，退至走廊，力图收集更多的传感器数据，从而在多假设信任度状态中增加了一个位置的相对似然值。

虽然计算上并不吸引人，但人们可以更进一步想象一下 Dervish 机器人的规划系统，它指定目标信任度状态而不是目标位置。为了达到目标置信度的水平，机器人可以推理和规划，因而不仅明确地考虑了机器人位置，而且考虑了各位置所量测到的似然值。这种过程的一个例子是 Latombe[141] 的传感器不确定场。在该例子中，机器人必须找到达到目标的轨迹，同时使它定位的置信度在线最大化。

5.6.2.3 实例研究 2：利用栅格图的马尔可夫定位

环境纯拓扑分解的主要缺点是由于这种粒度表示所造成的分辨率限制。在这种情况下，机器人的位置通常被限制为单节点的分辨率。对某些应用，这是不希望的。

在本实例研究中，我们检验一下 Burgard 和他的同事的研究工作[49,50]。他们用基于栅格的表示方法，实现很高精度的导航，同时仍使用马尔可夫定位技术。

该研究所使用的机器人 Rhino 是一个有 24 个声纳和 2 个 Sick 激光测距仪的 RWI B24 机器人。显然，在传感器级别上，该机器人有可能比装在 Dervish 上少量声纳传感器积累更多和更准确的距离数据。为了最大限度地利用这些细粒度传感器数据，Rhino 采用了占用和非占用空间的一个 2D 几何环境表示方法。这个计量地图有规则地被棋盘格划分成**固定分解**的栅格。在不同的示例中，栅格各单元占有 4～64 cm。

像 Dervish 一样，Rhino 使用多假设信任度表示方法。为了极大地改善环境表示分辨率，Rhino 的信任度状态表示由 15×15×15 3D 阵列组成，它表示 15^3 可能的机器人位置的概率(图 5.23)。阵列的分辨率是 15 cm×15 cm×1°。注意，与 Dervish 假定它的方向是近似的和已知的不同，Rhino 显式地以细粒度表示可选方向。所以，它的信任度状态形式上表示为 3 个自由度。正如我们以前所述那样，为了整个系统很好地运行，信任度状态表示的分辨率必须与环境表示相匹配。

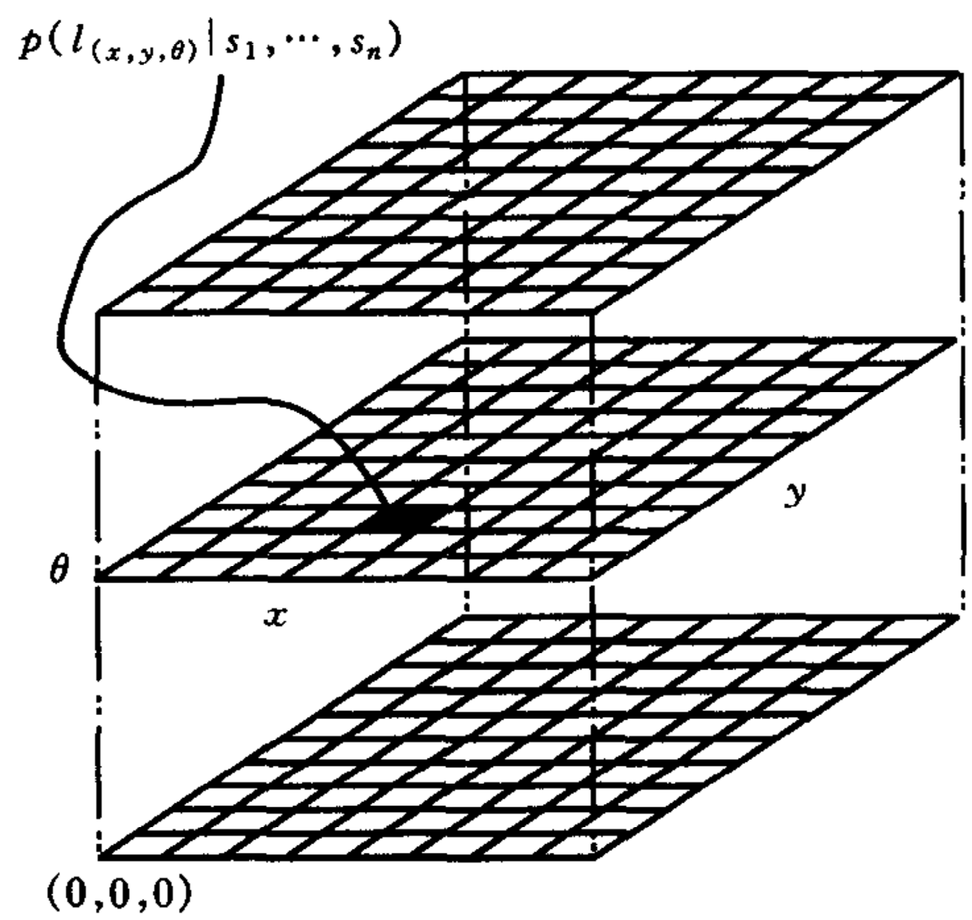


图 5.23 Rhino 所使用的信任度状态表示 3D 阵列(图片由 W. Burgard 和 S. Thrun 提供)

尽管 Dervish 只利用感知事件，忽略编码器输入，从而完全忽略了计量距离，Rhino 采用了 5.6.2.1 节概括的完全马尔可夫概率定

位方法。在每一个周期，既包括明确的动作更新阶段，又包括感知更新阶段。

由于位置的棋盘表示，所以执行了离散马尔可夫链型式的动作更新。给定在时刻 t 编码器测量 o，在信任度状态中，各更新的位置的概率表达为所有以前可能位置与运动模型之和：

$$P(l_t \mid o_t) = \sum_{l'} P(l_t \mid l'_{t-1}, o_t) \cdot p(l'_{t-1}) \tag{5.26}$$

注意到方程(5.26)只不过是方程(5.22)离散型式。Rhino 所用的特定的运动模型将运动结果表示成有界的高斯(即分布的尾部是有限的)。Rhino 的运动学结构是一个三轮同步驱动而不是差动驱动的机器人。然而，图 5.4 和图 5.5 所刻画的误差椭圆同 Rhino 运动模型产生的高斯界限是相同的。

感知模型精确地服从方程(5.21)的贝也斯方程。给定距离感知 i，机器人在各位置 l 的概率按下式更新：

$$p(l \mid i) = \frac{p(i \mid l)\, p(l)}{p(i)} \tag{5.27}$$

注意，虽然分母对变化的值 l 是恒定的，但它是一个被 Rhino 使用的分母。该分母作用如同一个规范器，保证信任度状态中的概率测量，其和恒定地为 1。

当然，主要的难题是计算 $p(i \mid l)$。在 Dervish 情况下，i 和 l 可能值的数目是如此地少，以至一个简单的表可以满足。然而用 Rhino 的细粒度计量表示，可能的传感器读数的数目和环境的几何上下关系是极多的。因此，Rhino 采用机器人传感器的行为模型、它的位置 l 和环绕 l 的局部环境计量地图，直接计算 $p(i \mid l)$。

假定感知测量的似然值被声纳或激光测距仪传感器的已知误差所证实，传感器模型必须计算特定感知测量的概率。为构建该传感器模型，使用以下三个关键的假设：

(1) 如果距离传感器检测到计量地图中的物体，则测量误差可以用分布予以描述，该分布在正确读数中有均值；

(2) 应该经常存在距离传感器读出任何测量值的非零机会，甚至该测量值与环境几何特征明显地不一致；

(3) 与(2)描述的一般误差相反，在测距传感器，存在一个特殊的失误模式。由此，信号被吸收或固有地被反射，致使传感器的距离测量值为最大。所以，在距离传感器的最大读数上存在概率密度分布的局部峰值。

在多个环境中，利用实验声纳的试验，证实了以上这些假设。研究小组已经对 Rhino 的特殊传感器，提出了一个保守且功能强大的传感器模型。

图 5.24 给出了一个简单的基于栅格的马尔可夫定位算法的 1D

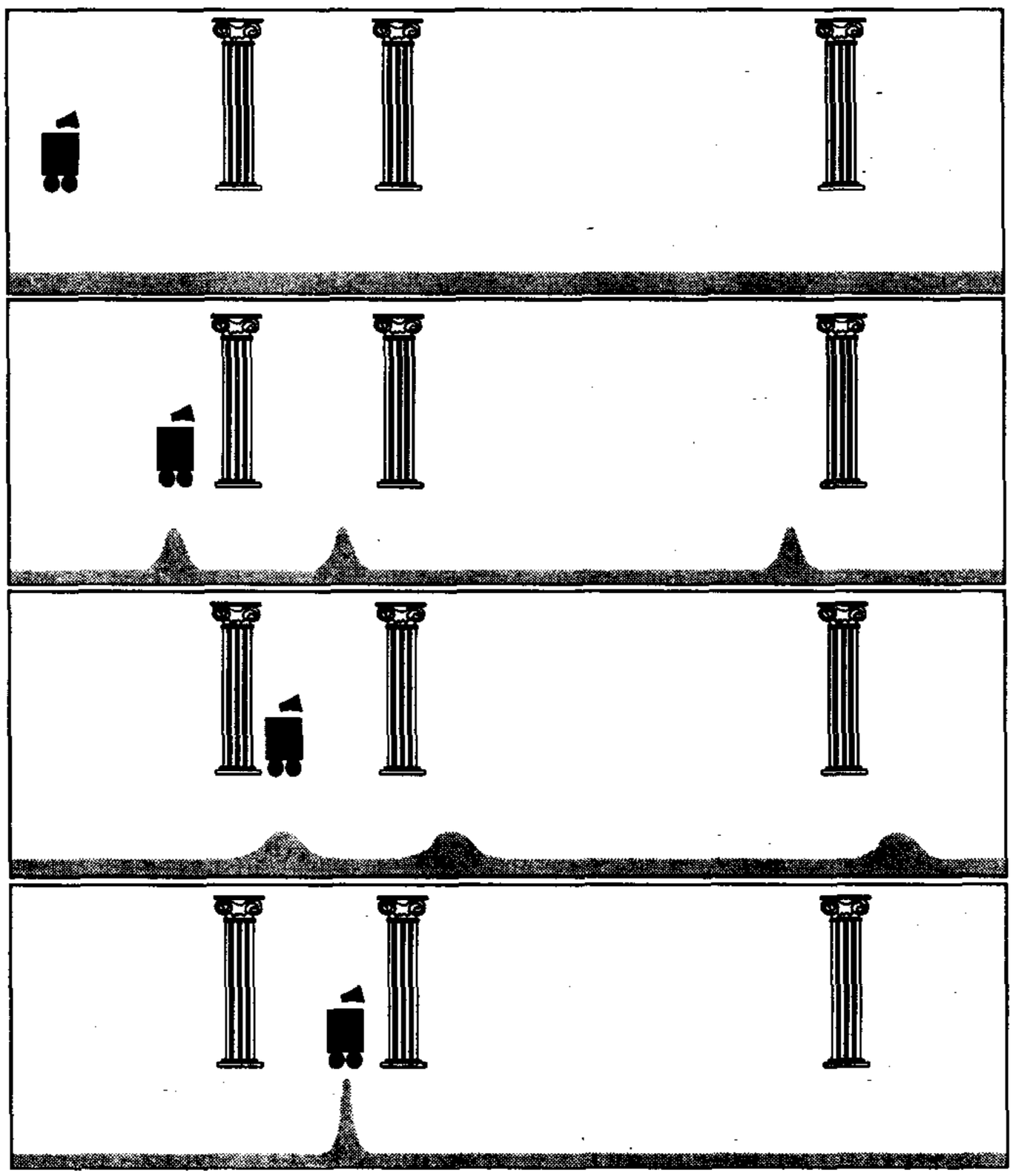


图 5.24 移动改善信任度状态

例子。机器人从它可能位置的平坦概率密度函数开始，换句话说，它一开始没有关于位置的偏差值，因为机器人首先碰到一个门，然后第二个门，对可能位置的概率密度函数先变成多模，最后为单模，且清晰地被确定。从一个初始失去信任度状态开始有能力定位机器人，是马尔可夫定位系统的重要特征。

最终形成的机器人定位系统已成为导航系统的一部分。它在波恩大学和波恩博物馆获得了很大的成功。因为环境的动态性质，这是一个具有挑战性的应用。由于人群围着机器人，机器人的传感器经常受到碰撞。Rhino 在这样的布局中能运行良好，是马尔可夫定位方法强有力的证明。

减少计算复杂性：随机采样 为了达到计算的增益，在现实世界实现 Rhino 中，采取了许多重大步骤。这些都是有价值的。因为采用精确的单元分解表示，并使用原传感器的值而不是对特征的抽象，所以，这样的机器人就有与各感知相关的大量的计算工作量。

有一类技术值得一提，因为它可以极大地缩减用固定单元的分解表示技术的计算开销。我们称为**随机采样**的这个基本概念，又叫做粒子滤波器算法、浓缩算法或蒙特卡洛算法[68,144]。

不考虑特殊技术，基本算法在所有情况下是一样的。**每个**可能机器人的位置，不必通过表示完全和正确的信任度状态来表示，而只表示必须考虑的可能位置全集的一个**子集**，以构建近似的信任度状态。

例如，在时刻 t，考虑一个具有 10,000 个可能位置的全信任度状态的机器人。我们不必基于新的传感器测量来跟踪和更新 10,000 可能位置，机器人可以只选已存储位置的 10%，且只更新这些位置。采样过程用位置的概率值加权，人们可以将系统进行偏置，在概率密度函数的局部峰值处产生更多的采样点。所以，最终的 1000 个位置将基本上集中在最高概率的位置。这样偏置是所期望的，但只对一个点。

我们还希望能保证跟踪某些较小可能的位置。否则，如果机器人确实收到不可能的传感器测量，定位就会失败。这个采样过程的**随机化**，可以加上从平坦分布中来的附加样本予以实现。这些随机化方法的进一步增强，譬如，根据正在进行中的系统定位的置信度，能使统计样本数目飞快地变化的。这进一步减少了平均所需的样本数目，同时保证在需要时[68]使用大量的样本。

这些采样技术，已经产生了功能与先前全信任度状态集合的机器人完全一样的机器人，但计算上只用资源的一小部分。当然，这种采样有一个缺陷：缺乏完备性。即这些采样技术破坏了马尔可夫定位概率上的完备性，因为机器人不能更新**全部**非零概率的位置。因而存在一个危险：由于一个未必不可能但是正确的传感器读数，机器人真的可能失败。当然，正如使用全马尔可夫定位技术一样，恢复失败状态是可行的。

5.6.3 卡尔曼滤波器定位

马尔可夫定位模型可以表示机器人位置的任何概率密度函数。该方法是很普通的，但由于它的通用性，所以效率低。让我们对机器人定位的关键需求，换一种考虑。人们可以认为，关键的需求不是概率密度曲线的精确复制，而是对鲁棒定位至关重要的**传感器融合**问题。机器人通常包含大量的异质的传感器，各提供关于机器人定位的消息。关键是，各传感器都经受着它自己的失误模式。最优定位应该考虑所有传感器提供的信息。本节我们描述实现该信息融合的一个高效技术，称之为卡尔曼滤波器。该机制实际上比马尔可夫定位更有效，因为如下所述，在表示机器人信任度状态的概率密度函数，甚至是它的单个传感器读数时，有重要简化。但这个简化的好处是得到一个**合成的最优回归数据处理算法**。它不管精度，整合了全部信息，对感兴趣的变量（即机器人位置）估计当前值。卡尔曼滤波器的一般介绍可参考文献[106]，参考文献[3]提供了更详细的处理

方法。

图2.25描述了卡尔曼滤波器估计的一般方案。图中，系统有一个控制信号和作为输入的系统误差源。测量装置能够测量带有误差的某些系统状态。卡尔曼滤波器是根据**系统**的知识和**测量装置**，产生系统状态最优估计的数学机制，是对系统噪声、测量误差和动态模型不确定性的描述。因此，卡尔曼滤波器以最优的方式**融合**传感器信号和系统知识。其最优性依赖于评估的特性指标和假设所选的判据。在卡尔曼滤波器理论中，系统假定为是**线性**并具有**白色高斯**噪声。如同以前我们已讨论过那样，对我们移动机器人应用来说，高斯误差的假定是无效的。尽管如此，其结果是极有用的。在其他工程学科，高斯误差假定在某些场合被证明是十分正确的[106]。

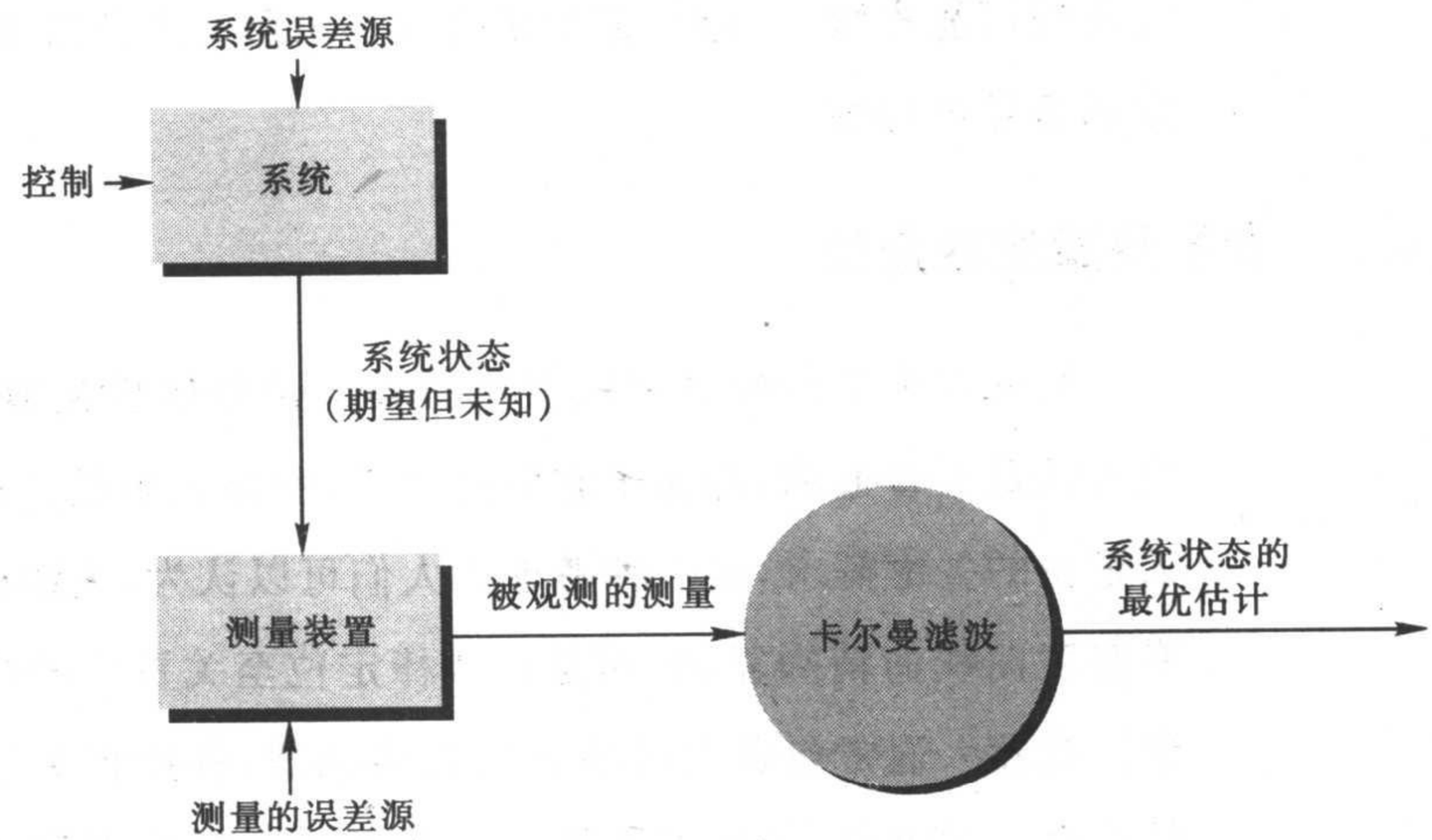


图5.25 典型的卡尔曼滤波器应用[106]

这一节以介绍卡尔曼滤波器理论开始，然后我们提出该理论应用于移动机器人的定位问题(5.6.3.2节)。最后，在5.6.3.2这一节中，根据卡尔曼滤波器定位，提出一个导航室内空间的移动机器人的实例。

5.6.3.1 卡尔曼滤波器理论介绍

基本的卡尔曼滤波器方法容许多个测量被优化地合并成对状态

的单一估计。为展示这一点，首先我们作简化假定，即在获取第 1 和第 2 测量之间，状态不变(例如，机器人不移动)。在提出这种静态情况之后，我们就容易引入**动态预测**。

静态估计 假定我们机器人有两个传感器：超声测距传感器和激光测距传感器。激光测距传感器为定位提供非常丰富和更准确的数据，但它经受与声纳测距仪不同的失误模式。例如，玻璃墙对激光将是透明的。但在迎面测量时，声纳会提供准确的读数。因而，我们希望联合由两个传感器提供的信息。但要知道，这样的传感器融合，按原理性方式进行时，只能产生信息的增益。

只要我们愿意用单模、零均和高斯噪声近似这些传感器的误差特征，卡尔曼滤波器就能使这种融合极其有效。特别地，假定我们已经获得两个测量，一个是在 k 时刻声纳传感器的测量；一个是在 $k+1$ 时刻激光测距仪的测量。根据各个测量，我们可以单独地估计机器人的位置。这样来自声纳的估计是 $\hat{q}_1$，基于激光的位置估计是 q_2。作为表征与这些估计的各有关的简化方法，我们预先假定一个(单模)高斯概率密度曲线，并由此把方差 σ_1^2 和 σ_2^2 与各测量关联起来。在图 5.26 中，虚线刻画了两个这种测量。归纳起来，得到两个

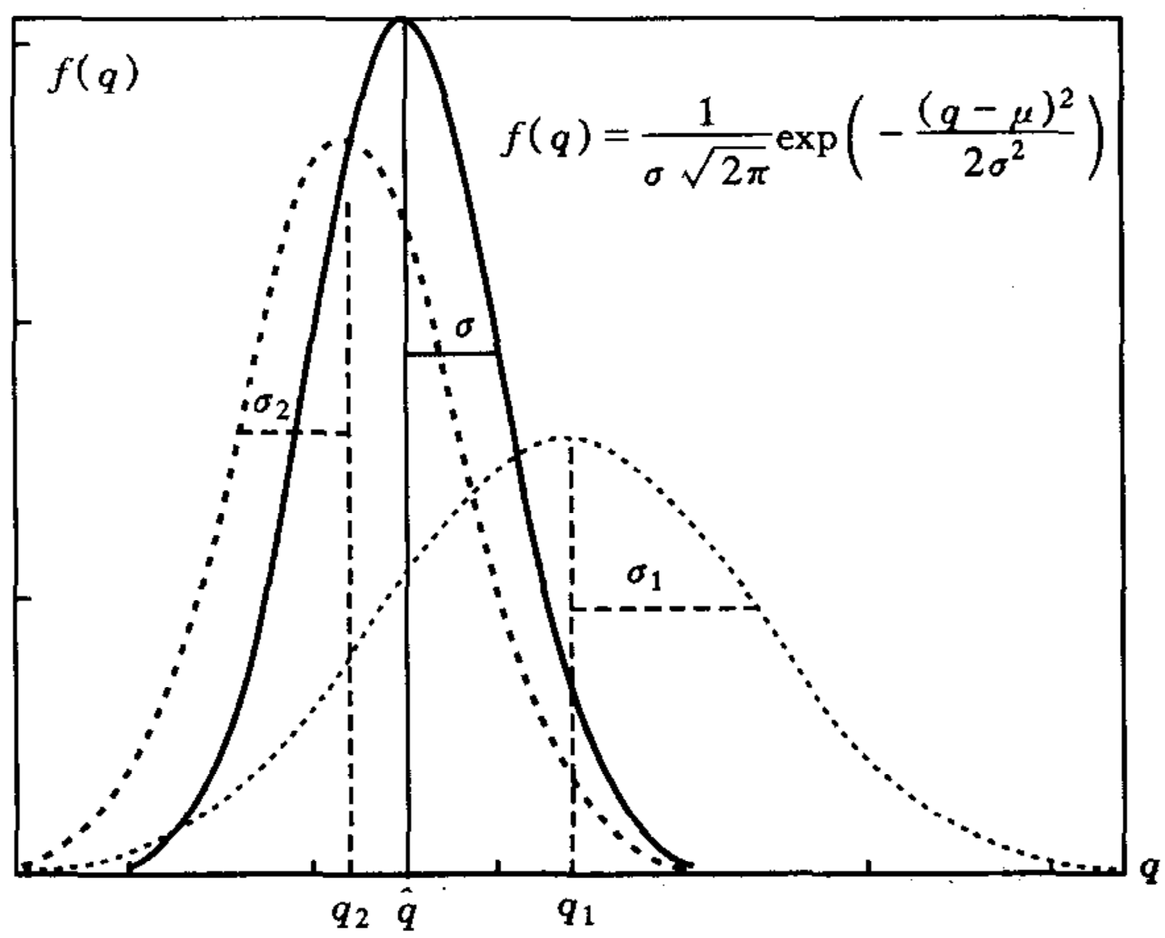


图 5.26 融合两个估计概率密度

机器人的位置估计：

$$\hat{q}_1 = q_1 \quad 具有方差\ \sigma_1^2 \tag{5.28}$$

$$\hat{q}_2 = q_2 \quad 具有方差\ \sigma_2^2 \tag{5.29}$$

问题是我们如何**融合**(联合)这些数据以得到机器人位置的最佳估计 $\hat{q}$? 我们假定在时刻 k 和 $k+1$ 之间机器人没有移动，这样我们可以直接应用 4.3.1.1 节方程(4.72*)相同的加权最小平方技术。因此我们写出：

$$S = \sum_{i=1}^{n} w_i (\hat{q} - q_i)^2 \tag{5.30}$$

式中 w_i 是测量 i 的权值。为了得到最小误差，我们使 S 的导数等于零。

$$\frac{\partial S}{\partial \hat{q}} = \frac{\partial}{\partial \hat{q}} \sum_{i=1}^{n} w_i (\hat{q} - q_i)^2 = 2 \sum_{i=1}^{n} w_i (\hat{q} - q_i) = 0 \tag{5.31}$$

$$\sum_{i=1}^{n} w_i \hat{q} - \sum_{i=1}^{n} w_i q_i = 0 \tag{5.32}$$

$$\hat{q} = \frac{\sum_{i=1}^{n} w_i q_i}{\sum_{i=1}^{n} w_i} \tag{5.33}$$

如果我们取权值 w_i 为

$$w_i = \frac{1}{\sigma_i^2} \tag{5.34}$$

那么，根据两个测量值，$\hat{q}$ 值可确定如下：

$$\hat{q} = \frac{\frac{1}{\sigma_1^2} q_1 + \frac{1}{\sigma_2^2} q_2}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} q_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} q_2 \tag{5.35}$$

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} = \frac{\sigma_2^2 + \sigma_1^2}{\sigma_1^2 \sigma_2^2}; \quad \sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_2^2 + \sigma_1^2} \tag{5.36}$$

注意，由方程(5.36)我们可以看到，合成的方差 σ^2 比单独测量

* 原书有误——译者注。

的所有方差 σ_i^2 都小。这样，位置估计的不确定性由于两个测量的联合已经减少。在图 5.26 中，实线概率密度曲线表示卡尔曼滤波器的结果，即说明了这个情况。即使测量不好，如声纳所提供的测量，只会增加估计的精度。这是基于信息论的我们所期待的结果。

方程(5.35)可以重写为：

$$\hat{q}=q_1+\frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}(q_2-q_1) \tag{5.37}$$

或者，按卡尔曼滤波器实现的最后形式，

$$\hat{x}_{k+1}=\hat{x}_k+K_{k+1}(z_{k+1}-\hat{x}_k) \tag{5.38}$$

式中，

$$K_{k+1}=\frac{\sigma_k^2}{\sigma_k^2+\sigma_z^2};\quad \sigma_k^2=\sigma_1^2;\quad \sigma_z^2=\sigma_2^2 \tag{5.39}$$

方程(5.38)告诉我们，在时刻 $k+1$，状态 x_{k+1} 的估计 $\hat{x}_{k+1}$ 等于在采集新测量 z_{k+1} 之前最佳预测值 $\hat{x}_k$，加上最优加权值校正项乘上时刻 $k+1$ 最佳预测值 $\hat{x}_k$ 和 z_{k+1} 之差。状态 $\hat{x}_{k+1}$ 的更新方差用方程(5.36)给定：

$$\sigma_{k+1}^2=\sigma_k^2-K_{k+1}\sigma_z^2 \tag{5.40}$$ *

由卡尔曼滤波器提供机器人位置新的融合的估计，再次受到高斯概率密度曲线的约束。它的平均值和方差是输入平均值和方差的简单函数。因此卡尔曼滤波器既提供了精密、简单的不确定性表示，又为联合异质的估计获取机器人位置新估计，提供一个极有效的技术。

动态估计 下面，考虑一个在相继的传感器测量之间有移动的机器人。假定在时刻 k 和 $k+1$ 之间，机器人的运动由速度 u 和代表实际速度不确定性的噪声进行描述：

$$\frac{\mathrm{d}x}{\mathrm{d}t}=u+w \tag{5.41}$$

* 原书有误——译者注。

如果现在我们在时刻 k 开始，知道机器人位置在该时刻的方差 σ_k^2 和运动的方差 σ_w^2，在刚才采集测量时刻 k'，我们得到

$$\hat{x}_{k'} = \hat{x}_k + u(t_{k+1} - t_k) \tag{5.42}$$

$$\sigma_{k'}^2 = \sigma_k^2 + \sigma_w^2[t_{k+1} - t_k] \tag{5.43}$$

式中

$t_{k'} = t_{k+1}$；

t_{k+1} 和 t_k 分别为时刻 $k+1$ 和 k 以秒为单位的时间。

因此，$\hat{x}_{k'}$ 是正好在时刻 $k+1$ 采集测量时的机器人位置最佳预测。它描述了位置误差的增长，直至采集到新的测量(图 5.27)。

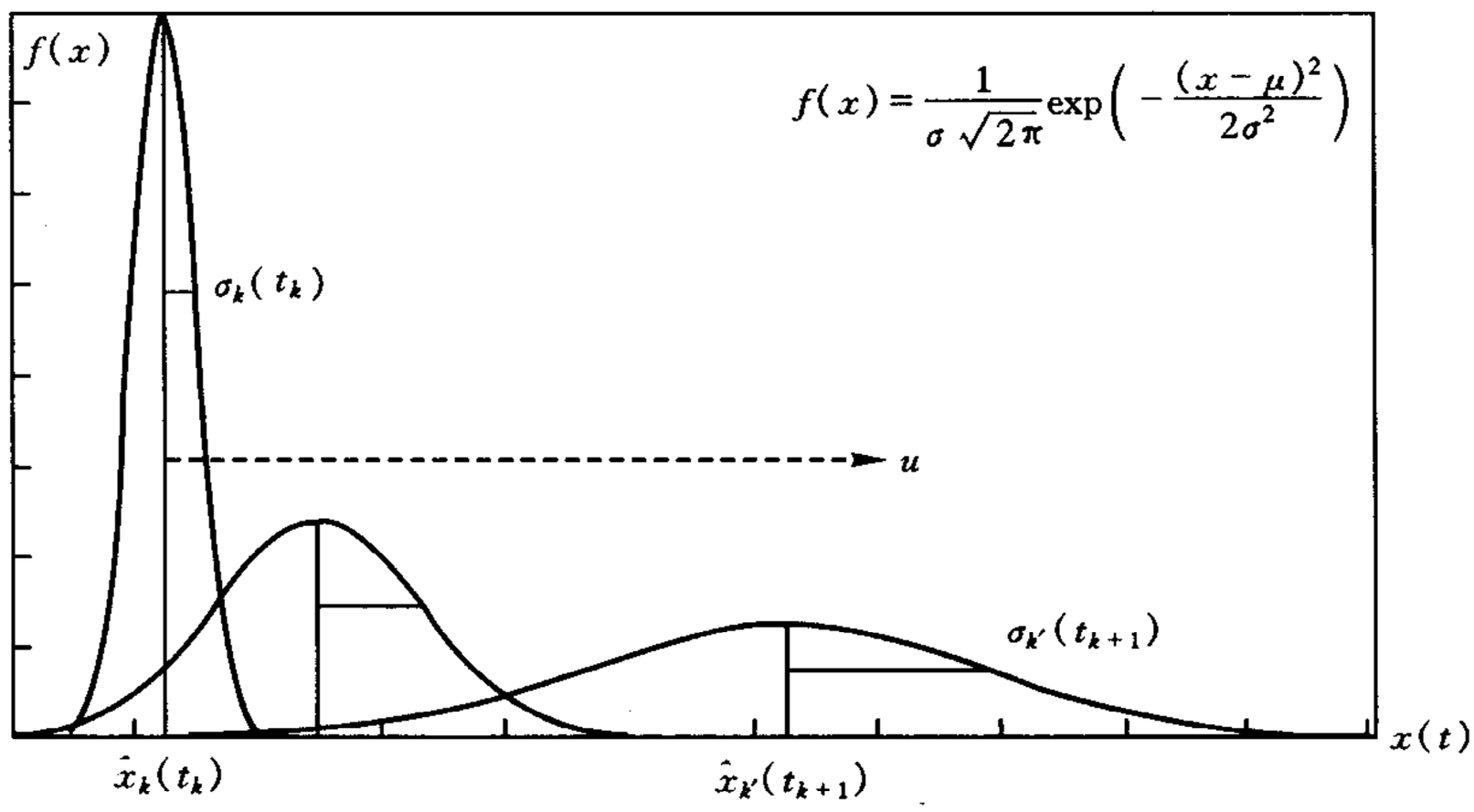


图 5.27 运动机器人概率密度的传播

现在我们可以用式(5.42)和(5.43)重写式(5.38)和(5.39)。

$$\begin{aligned}\hat{x}_{k+1} &= \hat{x}_{k'} + K_{k+1}(z_{k+1} - \hat{x}_{k'}) \\ &= [\hat{x}_k + u(t_{k+1} - t_k)] + K_{k+1}[z_{k+1} - \hat{x}_k - u(t_{k+1} - t_k)]\end{aligned} \tag{5.44}$$

$$K_{k+1} = \frac{\sigma_{k'}^2}{\sigma_{k'}^2 + \sigma_z^2} = \frac{\sigma_k^2 + \sigma_w^2[t_{k+1} - t_k]}{\sigma_k^2 + \sigma_w^2[t_{k+1} - t_k] + \sigma_z^2} \tag{5.45}$$

在时刻 $k+1$，最佳估计由时刻 k 上一次估计和机器人运动估计，包括所估计的运动误差来给出。

将上面方程扩展成向量情况，并考虑系统中时变参数和噪声描述，我们可以推导出卡尔曼滤波器定位算法。

5.6.3.2 应用于移动机器人：卡尔曼滤波器定位

卡尔曼滤波器是一个最优和有效的传感器融合技术。卡尔曼滤波器应用于定位要求将机器人定位问题陈述为传感器融合问题。回想机器人信任度状态的基本概率的更新，如方程(5.21)和(5.22)所指定的那样，可以被分成两个阶段：**感知更新**和**动作更新**。

卡尔曼滤波器方法和我们以前的马尔可夫定位方法之间的主要差别在于感知阶段。在马尔可夫定位中，整个感知，即机器人瞬时的传感器测量集合，是通过贝也斯方程单个地用于更新信任度状态中各可能的机器人位置。在某些情况下，感知是抽象的，已经由特征提取机制产生，如在 Dervish。在其他情况下，如 Rhino，感知由原始传感器读数组成。

相反，用卡尔曼滤波器的感知更新是一个多阶段过程。机器人的总传感器输入不是处理成单个整体，而是处理成所提取的特征集合，各特征与环境中物体有关。给定可能特征的集合，用卡尔曼滤波器融合各特征的距离估计，以匹配地图中的物体。马尔可夫方法对许多可能的机器人位置单个地实现这种匹配过程，与其不同的是卡尔曼滤波器一次性地通过处理全部单模和高斯的信任度状态，完成相同的概率更新。图 5.28 描述了卡尔曼滤波器定位的特殊架构。

第一步是动作更新或**位置预测**，把高斯误差运动模型直接应用于机器人所测量的编码器的行走。然后，机器人收集实际传感器数据，并在**观测**步骤中提取合适的特征(例如，直线、门，甚至指定传感器的值)。同时，根据地图中它的预测位置，机器人产生一个**测量预测**，它辨识机器人期待寻找的特征和那些特征的位置。在**匹配过程**中，机器人要在观测期间实际提取的特征和测量预测的期望特征之间，辨识最佳的配对。最后，卡尔曼滤波器可以将所有这些匹配所提供的信息融合，更新**估计**中机器人信任度状态。

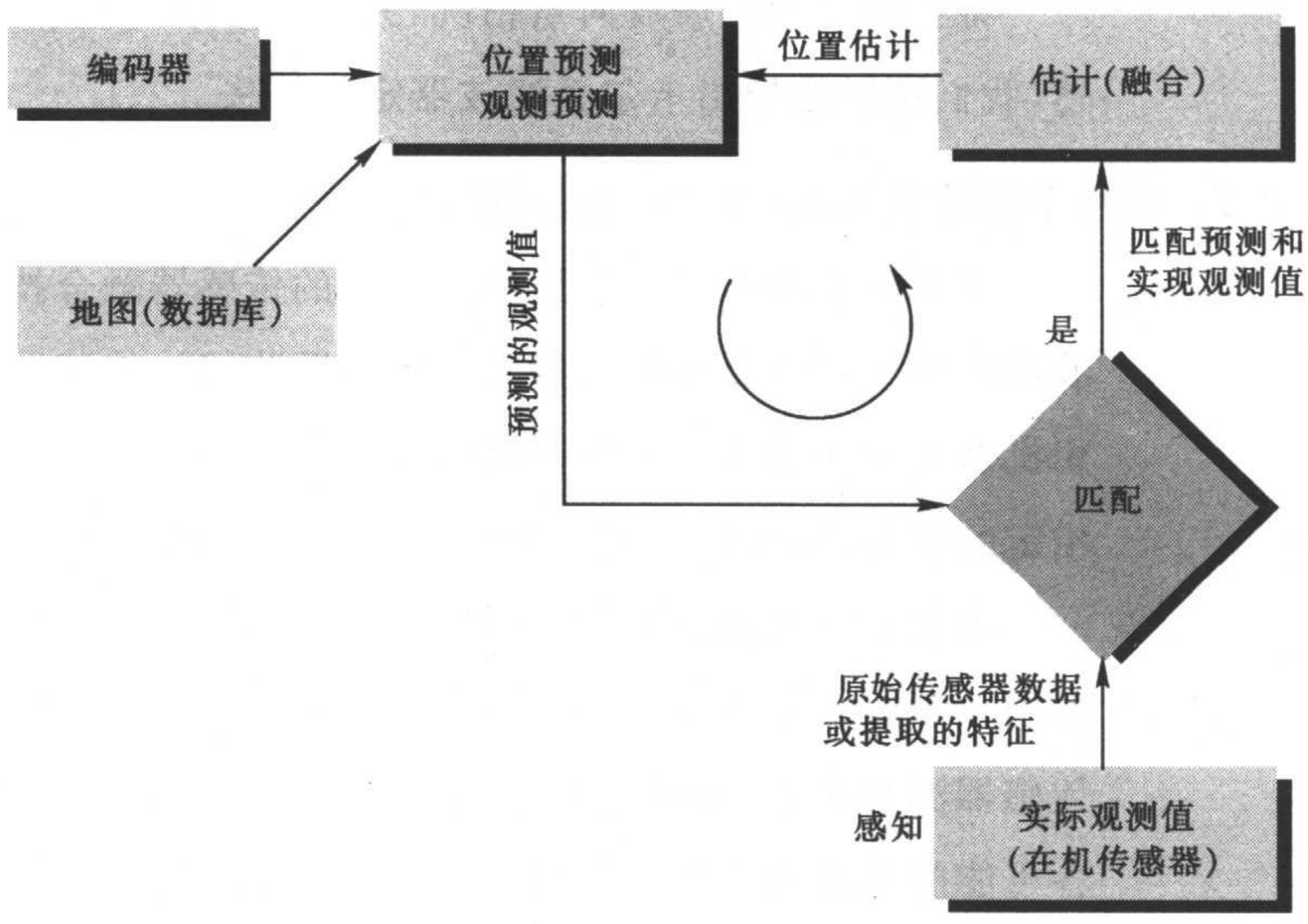


图 5.28 卡尔曼滤波器机器人定位的架构(见参考文献[23])

在以下各节我们更详细地描述这五个步骤。该陈述是基于 Leonard 和 Durrant-Whyte 的研究工作[23,pp.61-65]。

(1) **机器人位置预测** 根据它的老位置(在时间步 k)和由于控制输入 $u(k)$产生的运动,预测机器人在时间步 $k+1$ 的位置。

$$\hat{p}(k+1|k)=f(\hat{p}(k|k),u(k)) \tag{5.46}$$

对差动驱动的机器人,$\hat{p}(k+1|k)=p'$由方程(5.6)和(5.7)分别推导出来。

知道了过程和误差模型,我们还可以计算与该预测(见 5.2.4 节,方程(5.9))有关的方差 $\Sigma_p(k+1|k)$:

$$\Sigma_p(k+1|k)=\nabla_p f\cdot\Sigma_p(k|k)\cdot\nabla_p f^T+\nabla_u f\cdot\Sigma_u(k)\cdot\nabla_u f^T \tag{5.47}$$

这允许我们在由控制输入 $u(k)$确定的运动之后,预测机器人的位置和它的不确定性。注意到,信任度状态假定为高斯,所以,我们可以只用两个参数 $\hat{p}(k+1|k)$和 $\Sigma_p(k+1|k)$来表征信任度状态。

(2) **观测** 第二步,从时刻 $k+1$ 得到传感器测量 $Z(k+1)$。在

这表述中，我们假定，观测是在原传感器数据上执行特征提取的结果。所以，观测是由不同传感器提取的单独观测 $z_j(k+1)$ 的集合 n_0 所组成。形式上，各单独的观测可以代表一个所提取的特征，诸如一条直线或门，甚至一个单独的原始传感器的值。

特征的参数，通常是在传感器框架，即在机器人局部参考框架中指定。然而，对匹配而言，我们需要在相同的框架$\{S\}$中表示观测和预测。在我们表示的方法中，我们将把全局坐标框架的测量预测变换成传感器框架$\{S\}$。该变换在下一段落讨论的函数 h_i 中予以确定。

(3) **测量预测** 我们利用所预测的机器人位置 $\hat{p}(k+1|k)$ 和地图 $M(k)$ 产生多预测特征的观测 z_t，各预测的特征拥有变换到传感器框架的机器人位置：

$$\hat{z}_i(k+1)=h_i(z_t\hat{p}(k+1|k)) \tag{5.48}$$

我们可以把测量预测定义为包含所有 n_t 个预测特征的观测的集合：

$$\hat{Z}(k+1)=\{\hat{z}_i(k+1)|(1\leqslant i\leqslant n_t)\} \tag{5.49}$$

用预测状态估计 $\hat{p}(k+1|k)$ 来计算各预测的测量雅可比(Jacob)∇h_i，如同下面例子中将看到的那样，函数 h_i 主要是环境框架和传感器框架之间的坐标变换。

(4) **匹配** 在该点上，我们有实际、单独观测的集合，它是传感器空间的特征。我们还有预测到的特征集合，它也处于传感器空间中。匹配步骤具有辨识的作用，辨识与指定的预测特征匹配非常好的被用于估计过程的所有单个观测。换句话说，对一个观测子集和预测特征的子集，我们会找到配对，直觉上可以说“观测是根据地图其所预测特征的机器人测量值”。

形式上，匹配过程的目的，是产生一个从观测 $z_j(k+1)$ 到目标 z_t(存储在地图中)的分配。对相应于观测所找到的各测量预测，我们计算修正 $v_{ij}(k+1)$。**修正**是预测和观测之间差别的度量：

$$v_{ij}(k+1)=[z_j(k+1)-\hat{z}_i(k+1)]$$
$$=[z_j(k+1)-h_i(z_i\hat{p}(k+1|k))] \tag{5.50}$$

修正的协方差 $\Sigma_{IN,ij}(k+1)$ 可以用误差传播律(4.2.2 节,方程(4.60))求得:

$$\Sigma_{IN,ij}(k+1)=\nabla h_i \cdot \Sigma_p(k+1|k) \cdot \nabla h_i^T+\Sigma_{R,i}(k+1) \tag{5.51}$$

式中 $\Sigma_{R,i}(k+1)$ 表示测量 $z_j(k+1)$ 的协方差(噪声)。

为了确定测量预测和观测的对应的有效性,必须指定**确认门限**。确认门限的一个可能定义是 Mahalanobis 距离:

$$v_{ij}^T(k+1) \cdot \Sigma_{IN,ij}^{-1}(k+1) \cdot v_{ij}(k+1) \leqslant g^2 \tag{5.52}$$

然而,根据应用、传感器和环境模型可以使用更复杂的确认门限。对各预测的测量,用确认方程测试观测 $z_j(k+1)$ 在确认门限中的隶属度。当单个观测落入确认门限时,我们得到一个成功的匹配。如果,一个观测落入多个确认门限,则选择最佳的匹配候选解或跟踪多个假设。没有落入确认门限的观测,就被定位完全抛弃。这种观测可能是因物体不在地图中引起,诸如新的物体(例如,某人在走廊放了一个大箱子)或瞬时物体(例如,站在机器人旁边的人可形成一个直线特征)。一种办法是利用这种不匹配的观测来补充机器人的地图。

(5) **估计** 接下来根据位置预测和在时刻 $k+1$ 的所有观测,我们计算机器人位置的最佳估计 $\hat{p}(k+1|k+1)$。为了进行该位置的更新,我们首先把有效的观测 $\hat{z}_j(k+1)$ 叠加到单个向量上组成 $z(k+1)$,并指定复合修正 $v(k+1)$。然后,对各个有效的测量叠加测量的雅可比 ∇h_i,合在一起组成复合的雅可比 ∇h 和测量误差(噪声)向量 $\Sigma_R(k+1)=\mathrm{diag}[\Sigma_{R,i}(k+1)]$。于是我们可以根据方程(5.51)计算复合修正协方差 $\Sigma_{IN}(k+1)$。利用著名的结果[3],卡尔曼增益可以被写成为:

$$K(k+1)=\Sigma_p(k+1|k) \cdot \nabla h^T \cdot \Sigma_{IN}^{-1}(k+1) \tag{5.53}$$

我们可以更新机器人位置估计

$$\hat{p}(k+1|k+1)=\hat{p}(k+1|k)+K(k+1)\cdot v(k+1) \quad (5.54)$$

它具有相关的方差

$$\Sigma_p(k+1|k+1)=\Sigma_p(k+1|k)-K(k+1)\cdot\Sigma_{IN}(k+1)\cdot K^T(k+1) \quad (5.55)$$

对 1D 情况，并用 $h(z_p\hat{p}(k+1|k))=z_t$，我们可以证明这方程相当于以前推导的 1D 情况。

对应于式(5.45)，将式(5.53)简化成：

$$K(k+1)=\frac{\sigma_p^2(k+1|k)}{\sigma_{IN}^2(k+1)}=\frac{\sigma_p^2(k+1|k)}{\sigma_p^2(k+1|k)+\sigma_R^2(k+1)} \quad (5.56)$$

对应于式(5.44)，式(5.54)简化成：

$$\begin{aligned}\hat{p}(k+1|k+1)&=\hat{p}(k+1|k)+K(k+1)\cdot v(k+1)\\&=\hat{p}(k+1|k)+K(k+1)\cdot[z_j(k+1)\\&\quad-h_i(z_t\hat{p}(k+1|k))]\\&=\hat{p}(k+1|k)+K(k+1)\cdot[z_j(k+1j)-z_t]\end{aligned} \quad (5.57)$$

5.6.3.3 案例研究：具有特征提取的卡尔曼滤波器定位

EPFL 的皮格马利翁(Pygmalion)机器人是一个差动驱动的机器人，它使用激光测距仪作为它的主要传感器[37,38]。与 Dervish 和 Rhino 二者相反，皮格马利翁的环境表示是连续和抽象的：地图由无限的描述环境的直线组成。当然，皮格马利翁的信任度状态表示为高斯分布，因为该机器人使用了卡尔曼滤波器定位算法。它的平均位置 μ 的值被表示成高度精确的。需要时，皮格马利翁可以很高的精度定位。下面，我们提出皮格马利翁实现卡尔曼滤波器定位的五个步骤的详细内容。为了简单，我们假定传感器框架{S}等于机器人框架{R}。如无特定情况，全部的向量都在环境坐标系{W}中表示。

(1) **机器人位置预测** 在时间增量 k，机器人处在位置 $p(k)=[x(k)y(k)\theta(k)]^T$，它的最佳位置估计是 $\hat{p}(k|k)$。控制输入 $u(k)$把

机器人驱动到位置 $p(k+1)$(图 5.29)。在时间增量 $k+1$,机器人的

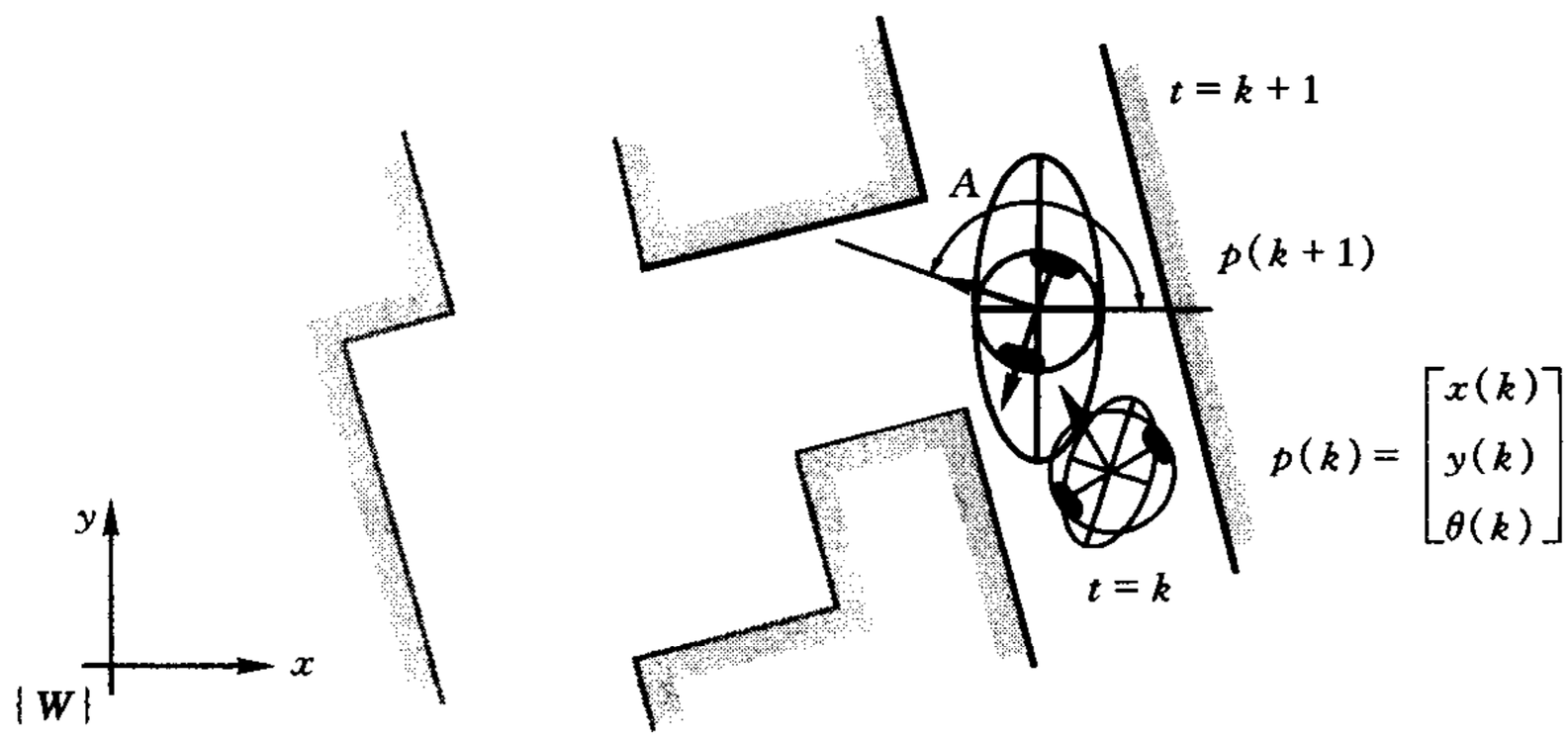


图 5.29 根据它的前一位置(细线)机器人的位置预测(粗线)和所执行的移动。围绕机器人所画的椭圆代表在 x,y 方向的不确定性(例如,3σ)。方向不确定性没有在图中标出

位置预测 $\hat{p}(k+1)$可以从以前的估计 $\hat{p}(k|k)$和里程表移动积分计算而得。对皮格马利翁所具备的差动驱动而言,我们可以用 5.2.4 节开发的模型:

$$\hat{p}(k+1|k)=\hat{p}(k|k)+u(k)=\hat{p}(k|k)+\begin{bmatrix}\frac{\Delta s_r+\Delta s_l}{2}\cos\left(\theta+\frac{\Delta s_r-\Delta s_l}{2b}\right)\\ \frac{\Delta s_r+\Delta s_l}{2}\sin\left(\theta+\frac{\Delta s_r-\Delta s_l}{2b}\right)\\ \frac{\Delta s_r-\Delta s_l}{b}\end{bmatrix} \tag{5.58}$$

和更新的协方差矩阵

$$\Sigma_p(k+1|k)=\nabla_p f\cdot\Sigma_p(k|k)\cdot\nabla_p f^T+\nabla_u f\cdot\Sigma_u(k)\cdot\nabla_u f^T \tag{5.59}$$

式中

$$\Sigma_u=\mathrm{cov}(\Delta s_r,\Delta s_l)=\begin{bmatrix}k_r|\Delta s_r| & 0\\ 0 & k_l|\Delta s_l|\end{bmatrix} \tag{5.60}$$

(2) **观测** 对于以直线为基础的定位,各个单独的观测(例如,

直线特征)是从原始激光测距仪读数中提取的,并分别由两个直线参数 $\beta_{0,j}$,$\beta_{1,j}$,或 α_j,r_j(图 4.36)组成。对一个旋转的激光测距仪,在极坐标框架内表示更为合适,所以这里我们用极坐标框架:

$$z_j(k+1)=\begin{bmatrix}\alpha_j\\ r_j\end{bmatrix}^R \tag{5.61}$$

在时刻 $k+1$,取得原数据之后,提取直线和不确定性(图 5.30(a)、5.30(b))。这就产生了具有 $2n_0$ 个直线参数(图 5.30(c))的 n_0

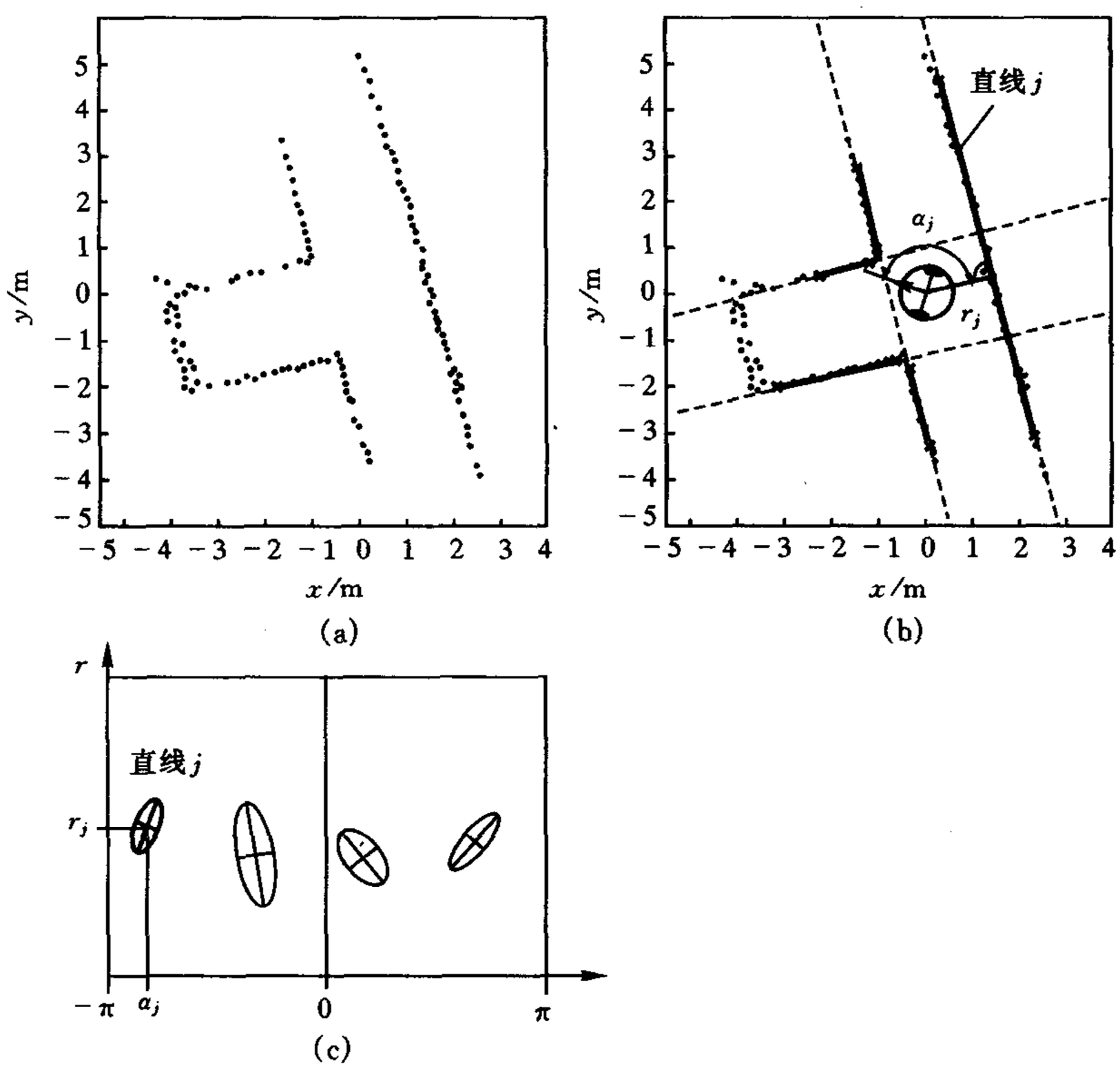


图 5.30 观测:在时刻 $k+1$ 从原始数据(a)由激光扫描仪获取,提取的直线(b);可以在模型空间中表示直线参数 α_j 和 r_j 和它的不确定性(c)

条的观测直线和各条直线的协方差矩阵。如在 4.3.1.1 节开发的直线提取所做那样,矩阵可以从所有贡献给各直线的测量点的不确定性计算而得:

$$\Sigma_{R,f}=\begin{bmatrix}\sigma_{\alpha\alpha} & \sigma_{\alpha r}\\ \sigma_{r\alpha} & \sigma_{rr}\end{bmatrix}_j \tag{5.62}$$

(3) **测量预测** 根据存储的地图和预测的机器人位置 $\hat{p}(k|k)$，产生期望特征 $z_{t,i}$ 的测量预测(图 5.31)。为了缩减所需的计算量，常常有一个附加步骤，它首先从地图特征全集中，选择可能的特征，在现在情况下，是直线。这些直线存储在地图中，并在环境坐标系[W]中指定。所以它们需要变换到机器人框架{R}：

$$W_{z_{t,i}}={}^{W}\begin{bmatrix}\alpha_{t,i}\\ r_{t,i}\end{bmatrix}\rightarrow R_{z_{t,i}}={}^{R}\begin{bmatrix}\alpha_{t,i}\\ r_{t,i}\end{bmatrix} \tag{5.63}$$

按照图(5.31)，变换由下式给定：

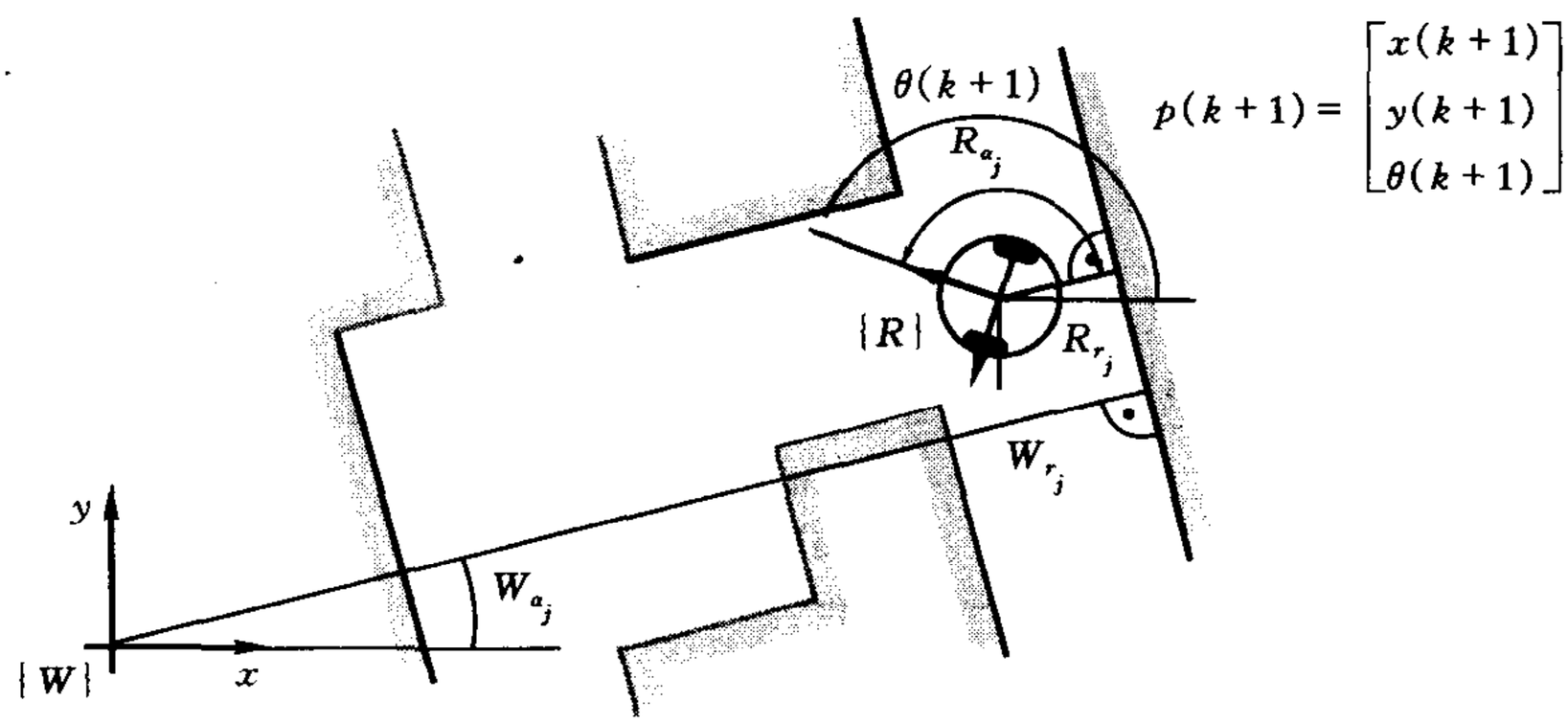


图 5.31 环境坐标系{W}和机器人坐标框架{R}中目标位置的表示

$$\hat{z}_i(k+1)={}^{R}\begin{bmatrix}\alpha_{t,i}\\ r_{t,i}\end{bmatrix}=h_i(z_{t,i},\hat{p}(k+1|k))$$
$$=\begin{bmatrix}{}^{W}\alpha_{t,i}-{}^{W}\hat{\theta}(k+1|k)\\ {}^{W}r_{t,i}-({}^{W}\hat{x}(k+1|k)\cos({}^{W}\alpha_{t,i})+{}^{W}\hat{y}(k+1|k)\sin({}^{W}\alpha_{t,i}))\end{bmatrix} \tag{5.64}$$

它的雅可比 ∇h_i 由下式给定：

$$\nabla h_j = \begin{bmatrix} \frac{\partial \alpha_{t,i}}{\partial \hat{x}} & \frac{\partial \alpha_{t,i}}{\partial \hat{y}} & \frac{\partial \alpha_{t,i}}{\partial \hat{\theta}} \\ \frac{\partial r_{t,i}}{\partial \hat{x}} & \frac{\partial r_{t,i}}{\partial \hat{y}} & \frac{\partial r_{t,i}}{\partial \hat{\theta}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ -\cos^W \alpha_{t,i} & -\sin^W \alpha_{t,i} & 0 \end{bmatrix} \tag{5.65}$$

测量预测产生了在机器人坐标框架中所表示的预测直线(图5.32)。它们是不确定的,因为机器人位置预测是不确定的。

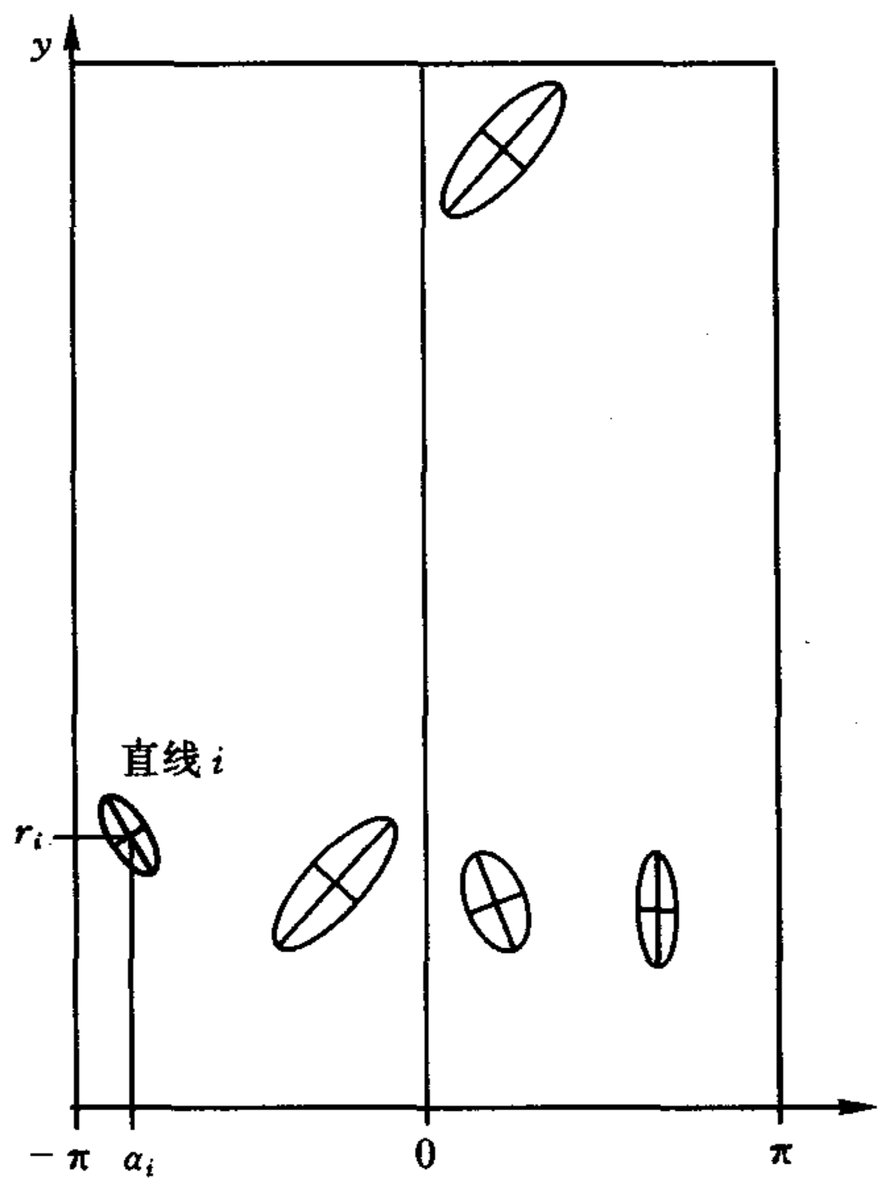


图 5.32 测量预测:根据地图和估计的机器人位置,预测目标(可见直线)。与观测一样,它们被表示在模型空间中

(4) **匹配** 为了匹配,我们必须在预测的和观测的特征(图5.33)之间找到对应(配对)。在我们的情况下,我们采用 Mahalanobis 距离:

$$v_{ij}^T(k+1) \cdot \Sigma_{IN,ij}^{-1}(k+1) \cdot v_{ij}(k+1) \leqslant g^2 \tag{5.66}$$

和

$$v_{ij}(k+1) = [z_j(k+1) - h_i(z_t, \hat{p}(k+1|k)]$$

$$= \begin{bmatrix} \alpha_j \\ r_j \end{bmatrix} - \begin{bmatrix} {}^W\alpha_{t,i} - {}^W\hat{\theta}(k+1|k) \\ {}^Wr_{t,i} - ({}^W\hat{x}(k+1|k)\cos({}^W\alpha_{t,i}) + {}^W\hat{y}(k+1|k)\sin({}^W\alpha_{t,i})) \end{bmatrix} \tag{5.67}$$

$$\Sigma_{IN,ij}(k+1) = \nabla h_j \cdot \Sigma_p(k+1|k) \cdot \nabla h_i^T + \Sigma_{R,i}(k+1) \tag{5.68}$$

以能够找到最佳的匹配，同时消除所有其他遗留的观测和预测不匹配的特征。

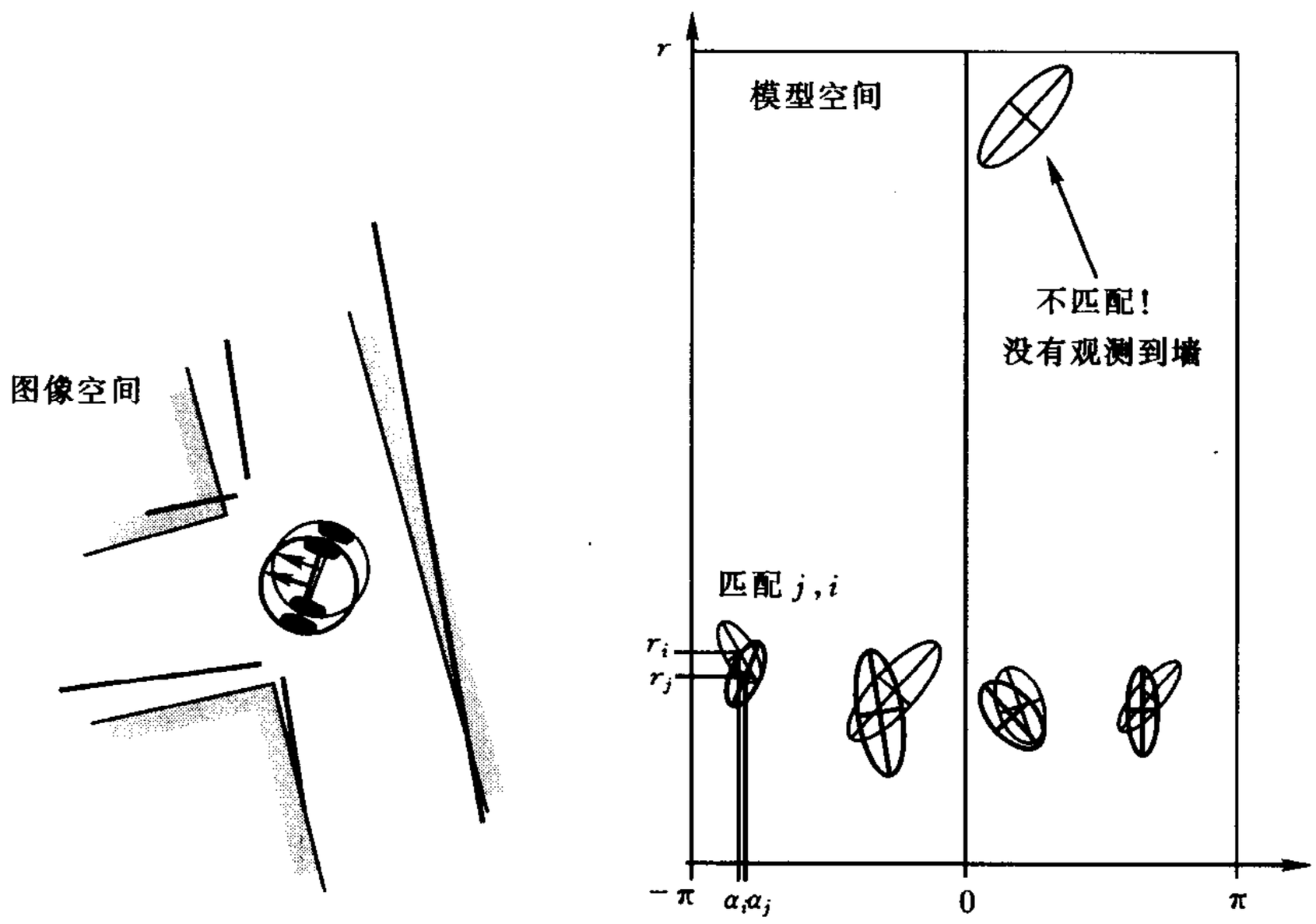


图 5.33 匹配：匹配观测（粗线）和测量（细线），计算修正和它的不确定性

(5) **估计** 应用卡尔曼滤波器产生最后的相应于加权和的姿态估计（图 5.34）：

- 各观测的和预测的特征匹配对的姿态估计；
- 基于里程表和观测位置的机器人位置估计。

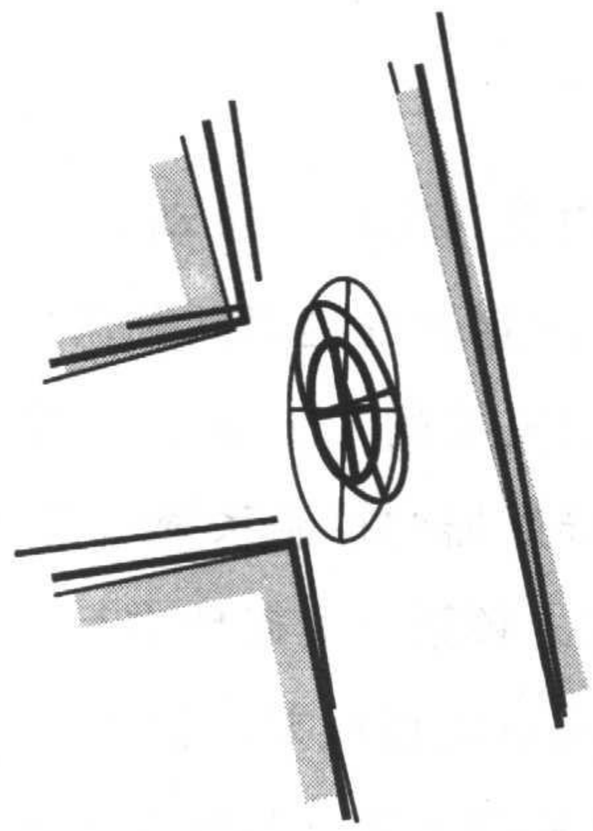


图 5.34 新机器人位置的卡尔曼滤波器估计：将机器人位置预测（细线）与测量（粗线）所得的修正融合，我们得到机器人位置（很粗线）的更新估计 $\hat{p}(k|k)$

5.7 定位系统的其他例子

马尔可夫定位和卡尔曼滤波器定位一直是研究移动机器人导航室内环境的两个极普遍的策略。它们有坚实的形式基础，所以有良定的行为特性。但是，在商业的和研究的移动机器人平台中，一直以不同的成功率，应用着许许多多其他定位技术。我们不打算详细探索所有定位系统的空间，读者可参考这种信息的综述，如参考文献[5]。

然而，有几个值得提出的定位技术范畴。不必惊奇的是，在商业机器人学中，这些技术的许多实现方法运用了机器人环境的改造。这是马尔可夫定位和卡尔曼滤波器定位团体所回避的某些问题。在以下各节，我们具体结合各范畴和参考的实例系统，适当时，包括那些改造环境和那些无环境改造要求的系统，简单地认识一下一般的策略。

5.7.1 基于路标的导航

路标一般定义为环境中被动的物体，当它们处在机器人视野之内时，提供了高的定位准确度。利用路标作定位的机器人，通常使用机器人设计者已安放的人工标记，使定位简易。

基于路标导航器的控制系统由两个分离的阶段组成。当看到一个路标时，机器人频繁且准确地进行定位，利用动作更新和感知更新跟踪它的位置而不累加误差。但当机器人处在无路标“区”时，只发生动作更新，且机器人累计位置的不确定性，直到下一个路标进入机器人的视野。

因此，机器人从一个路标区到另一个路标区有效地进行**航位测定**。这轮过来意味着机器人必须谨慎地咨询它的地图，以保证路标之间的各个移动足够地短。给定它的运动模型，它能够成功地定位到达下一个路标。

图5.35表示了基于路标导航的一个实例。路标的特殊式样能使机器人进行可靠准确的姿态估计，但利用路标间的**航位测定**，机器人必须行走。

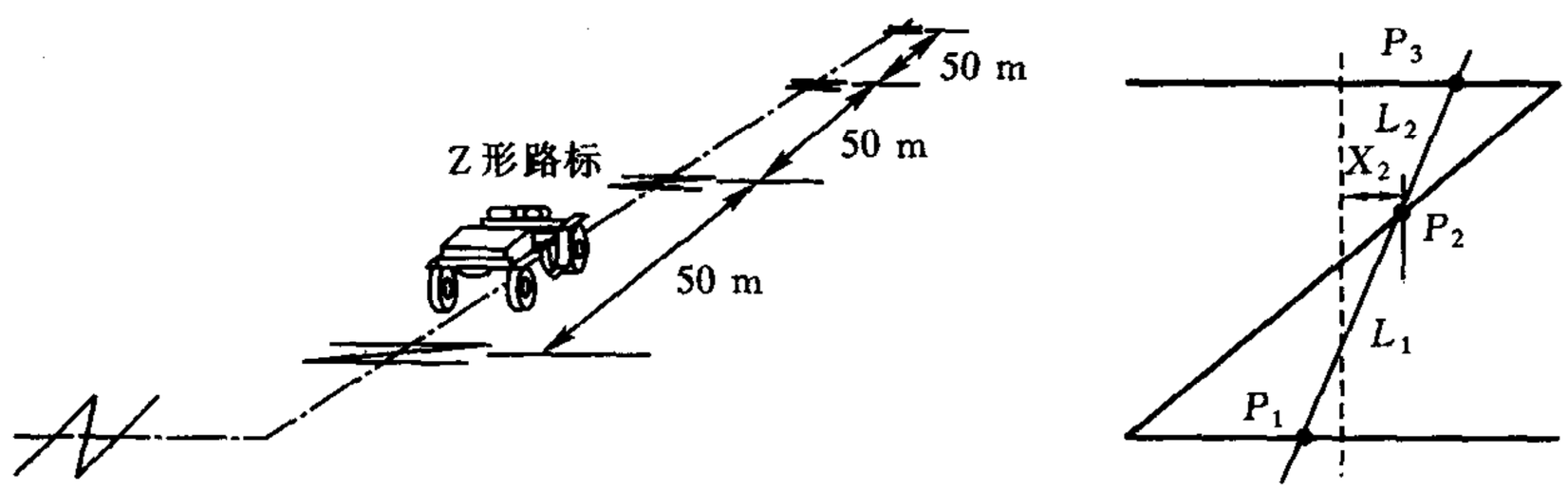


图5.35 地上的Z形路标，日本小松公司[5 pp. 179－180]

基于路标定位方法的一个主要优点是对这一般系统的结构[98]已经发展了坚实的形式理论。在这工作中，作者已经说明了精确的假设和条件。当满足假设和条件时，保证机器人能够经常地成功定位。这个研究也产生了一个基于路标定位的真实环境的示范系统。

在斯坦福大学机器人学实验室的天花板上安放了标准纸样，各有一个独特的方格图案。一个 Nomadics 200 的移动机器人装有垂直朝向天花板的单色 CCD 摄像机。通过识别放在 2 m 远的纸状路标，机器人能够定位到几个厘米的范围，然后利用航位测定法，移动到另一个路标区。

基于路标导航的基本缺点是，一般它要求重大的环境改造。路标是局部的，所以常常需要大量的路标以覆盖一个大的厂区或研究实验室。例如，斯坦福大学的机器人学实验室使用了约 30 个分离的路标，全部各自地粘贴在天花板上。

5.7.2 全局唯一定位

基于路标的导航，作了一个强的一般假定：当路标处在机器人的视野内，定位基本上是理想的。移动机器人达到**自如定位**的一个方法是，不管机器人在何处，都要有力地使这种假设有效。如果看一下机器人传感器，能马上唯一且重复地识别它的特殊位置，这将是革命性的。

这种定位策略确实是大胆且积极进取的，但它是否可以完成，本质上是一个传感器技术和感知软件的问题。显然，这样一个定位系统需要使用能收集极大量信息的传感器。因为视觉确实比以前的传感器收集远多得多的信息，所以在朝向全局唯一定位的研究中，它被选作为传感器。

图 4.49 描述了由反射折射摄像机系统所采集的图像。要是人类能注视一个像图片那样的个体并识别在众所周知环境中机器人的位置，那么人们可以断言，在图片内一定存在全局唯一定位的信息，只要弄清就是。

好几位研究者一直致力于这样的方法，并参与构建一个或多个直方图以稳定地表示图像的信息内容（例如，图 4.50 和 4.3.2.2 节）。使用这种图像-直方图系统的机器人，已经出现并用于在办公

楼中唯一地识别各个房间，以及在室外环境中识别各条人行道。然而这种系统对外部照明高度敏感，而且只提供与摄像机光学能见范围相当的定位分辨率水平。

在前一章，图4.39中表示的角度直方图是另一个例子。在该例子中，机器人的传感器读数被变换成位置的标识符。然而，由于声纳测距的有限的信息内容，在机器人环境中，有可能存在两个位置难以成功区分的太相似的角度直方图。

为全局定位聚集足够声纳信息的一个办法是，首先允许机器人定时将大量声纳信息聚集到局部证据栅格（即占用栅格）中，然后将局部证据栅格与环境全局的计量地图相匹配。在参考文献[129]中，研究工作者演示了这种系统，甚至当环境有重大变化时，系统也能立刻进行定位，降低了对地图的保真度要求。最有意义的是，局部证据栅格很好地表示了信息，它可以随时用于校正和更新地图。由此产生了向环境表示直接提供校正反馈的定位系统。性质上，这与卡尔曼滤波器定位算法中取得被拒绝的观测特征并用它们在地图中创建新特征的思想类似。

一个最有前途的、全局唯一定位的新方法称为**基于镶嵌图案(Mosaic)的定位**[83]。这个迷人的方法利用一个移动机器人罕用的环境特征：细粒度的地板纹理。这个方法之所以成功主要是因为最近普遍存在的超高速处理机、超高速摄像机和超大的存储介质。

机器人装有指向地板的高质量高速的CCD摄像机。理想地放在机器人轮子之间，并由一个远离摄像机轴的特殊光模进行照明，以增强地板纹理。机器人利用这个摄像机，从搜集机器人工作空间中整个地板的图像开始，进行工作。当然，存储器要求是巨大的，为了存储300×300面积的完整的图像库，需要10G的存储器。

一旦存储了完整的图像镶嵌图案，机器人可以在地板上行走任何轨迹，同时无困难地跟踪它自己的位置。定位实现只是记录一个图像，执行动作更新，然后通过基于图像数据库匹配的简单技术，将

图像与镶嵌图案进行匹配，执行感知更新。最终的特性指标令人印象深刻：这样的机器人已被证明，在以 25 km/h 的速度移动的同时，重复定位具有 1 mm 精度。

全局唯一定位的主要优点是：当这些系统正确运行时，它们大大地简化了机器人的导航。机器人可以移动到任何点上，而且通过收集传感器的扫描数据经常确保定位。

但是，该方法的主要缺点是：该方法可能从不提供定位问题的完全解。常常有这样情况，即局部传感器的信息确是含糊的。所以，只利用当前传感器信息，全局唯一定位是不可能成功的。人类通常有极好的**局部定位系统**，特别在非重复的和充分知晓的环境中，如在他们的家里。但是，有许多环境，例如，考虑一下树篱迷宫和大型新的、具有重复相同大厅的办公楼，即使对人类而言，这种即刻定位也是困难的。的确，上述基于镶嵌图案的定位原型在它一开始实现时就碰到这种问题。工厂车间的地面已被刚刚漆过，因而完全没有足够的细微裂痕来产生相关的纹理。他们的解决方案竟然是改造环境，随机地在工厂地面画上纹理。

5.7.3 定位信标系统

定位问题最可靠的解决方案之一是特别为目标环境设计和安装一个有源信标系统。这是工业和军事两个应用领域所使用的首选技术，是保证定位可靠性最大可能的一种途径。可以认为 GPS 系统正是这样一个系统(见 4.1.5.1 节)。

图 5.36 描述了机器人群的这种信标布局。正如用 GPS 一样，通过设计一个系统，机器人被动地定位，而信标是主动的，任意数目的机器人可同时利用单个的信标系统。如同大多数信标系统一样，其所描述的结构首先依赖于有效定位的几何原理。在该情况下为了将它们自己定位于全局坐标框架，机器人必须知道在该框架中两个有源超声信标的位置。

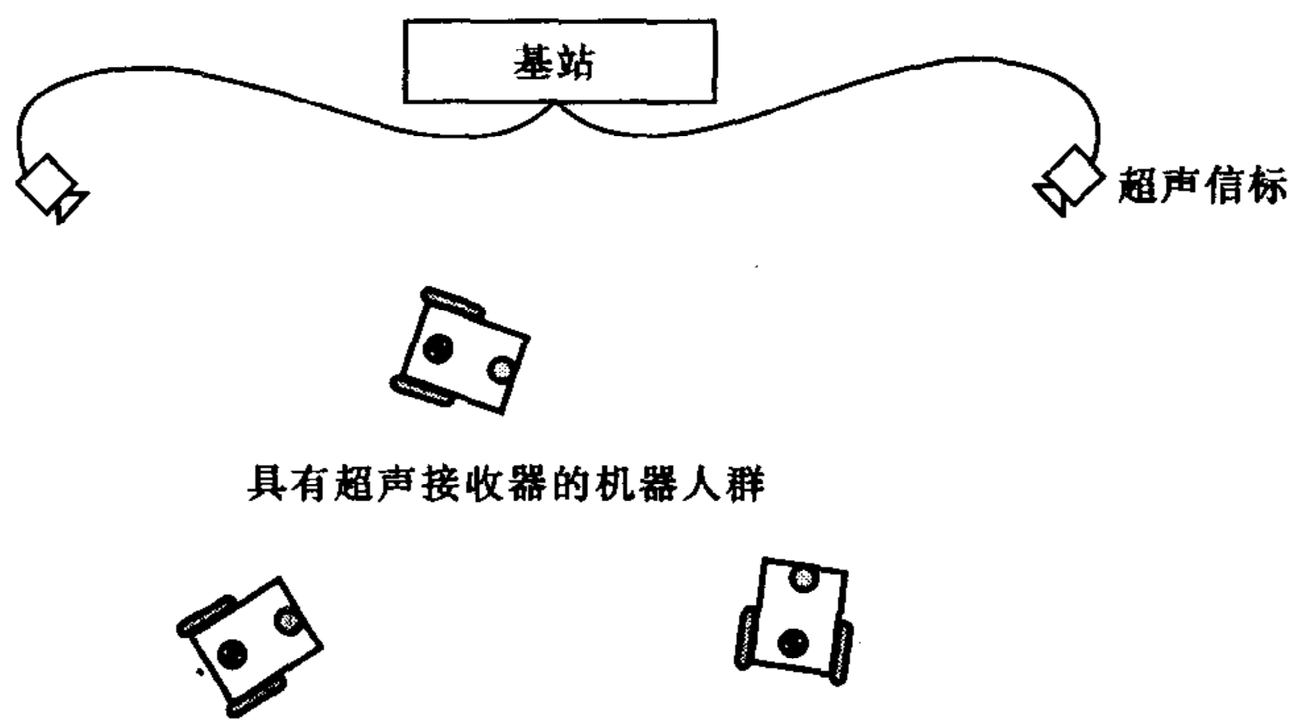


图 5.36 有源超声信标

在图 5.37 中，表示了工业机器人应用中一个流行型的信标系统。在此情况下，信标成为回射的标记，根据它们返回到机器人的反射能量，移动机器人容易检测到该标记。给定已知的光学回射器的位置，只要移动机器人有三个这种同时被看到的信标，它就可辨识它的位置。当然，具有编码器的机器人也可以随时定位，而且在同一时刻不需要测量相对于所有三个信标的角度。

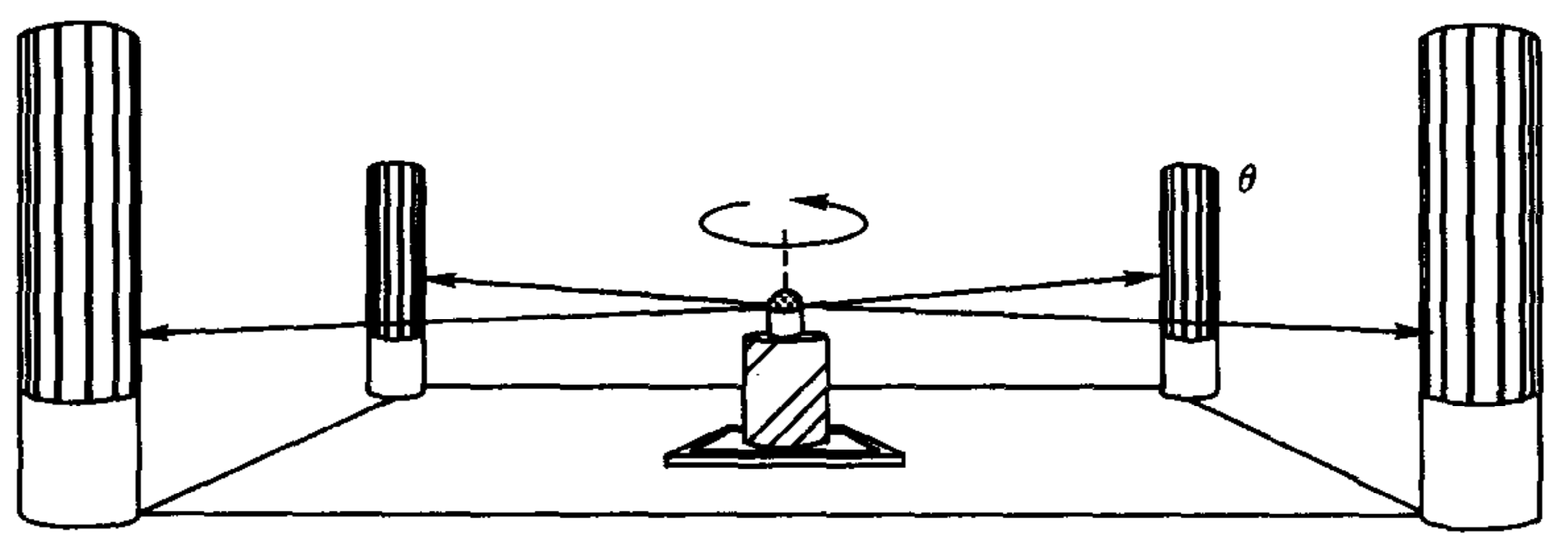


图 5.37 无源光学信标

这种基于信标系统的优点通常是极高的工程可靠性。利用相同的标记，在特定的商业环境中，重要工程的周围常常安装这种系统。所以，把机器人移动到不同的工厂车间，既费时又昂贵。通常，即使改变机器人所用的路由，也会需要重大的再造工程。

5.7.4 基于路由的定位

比基于信标系统更为可靠的是基于路由的定位策略。在这种情况下，机器人的路由被明显地标出，所以它能确定它的位置，允许机器人不相对于某全局坐标框架，而相对于指定的路径行走。标记这样的路由和相继的交叉口有许多技术。其中之一个是有效地建立一个铁路系统，不同的是该铁路系统比实际的铁路更灵活些，当然对人更友好。例如，高紫外反射、光学上透明的画可以标记路由，使得只有机器人用特殊的传感器才能容易地检测到它。另一种办法利用装在机器人盒子里的感应线圈，可以检测埋在大厅地下的导向线。

在所有这些情况下，机器人的定位问题，通过迫使机器人一直跟随一个预先指定的路径而有效地被简化。公正地说，现在有许多工业上的**无人导向车**为了壁障，暂时地偏离它们的路由。但这种极端可靠性的代价是显而易见的：使用这种定位方法，机器人更加不灵活。所以，机器人行为的任何变化，都需要重大的工程和时间。

5.8 自主地图的构建

我们已经讨论的所有定位策略都要求用人力把机器人安装到一个空间。人工的环境改造也许是必需的。即使不是这样，也必须为机器人构建一张环境地图。但是成功导航的机器人具有正确且合适的传感器。所以，机器人应该构建它自己的地图。这种企求追根到自主移动机器人学的核心。粗略地，可以把我们的最终目标表述如下：

从一个任意的初始位置开始，一个移动机器人应该能够用它随带的传感器探索环境，获得有关环境的知识，解释场景，构建合适的地图，并相对于地图将自己定位。

要稳健地实现这个目标可能还需要很多年，但一个重要的子目标是：创造环境地图的自主建立和修改的技术。当然，移动机器人的传感器只具有有限的量程，所以它必须在物理上探索它的环境以构建这种地图。因此，机器人不仅必须构建地图，而且在移动和定位以探索环境时，它也必须这样做。在机器人学的团体中，常常把它叫做同时的定位和作图(simultaneous localization and mapping，SLAM)问题。肯定地说，特别对移动机器人系统，这是最困难问题。

SLAM困难的原因，精确地说，来源于当机器人定位时它的位置更新和它的作图动作之间的交互。如果移动机器人根据不精确知道的特征的观测更新它的位置，其最终的位置估计就变得与特征位置估计相关。相似地，如果从不精确知道的位置取得观测，用于更新或在地图加一个特征，则地图就变得与位置估计相关。因此，地图构建的一般问题就是鸡和蛋问题的一个例子。为了定位，机器人需要知道特征在哪里；而为了构建地图，机器人需要知道地图上它在何处。

这个联合问题完整且最优解决的唯一途径是全面考虑位置估计和特征位置估计的相关性。这种互相关的地图被称为**随机地图**。下面一节[55]我们从讨论支持该方法的理论开始。

遗憾的是，实现这种最优解决方案计算上是禁忌的。为此，许多研究人员已经提出其他解决方法，这些方法在有限的环境中运行良好。5.8.2节表征了这些另选方法的部分解答。

5.8.1 随机构图的技术

图5.38表示了一个通用的架构，它把地图构建和维护，具体化为讨论卡尔曼滤波器定位[23]时由图5.28所描述的标准定位回路。所加的弧，代表在观测和测量预测之间不理想匹配时所发生的附加信息流。

非期望的观测会影响地图中新特征的创建，而不可观测的测量

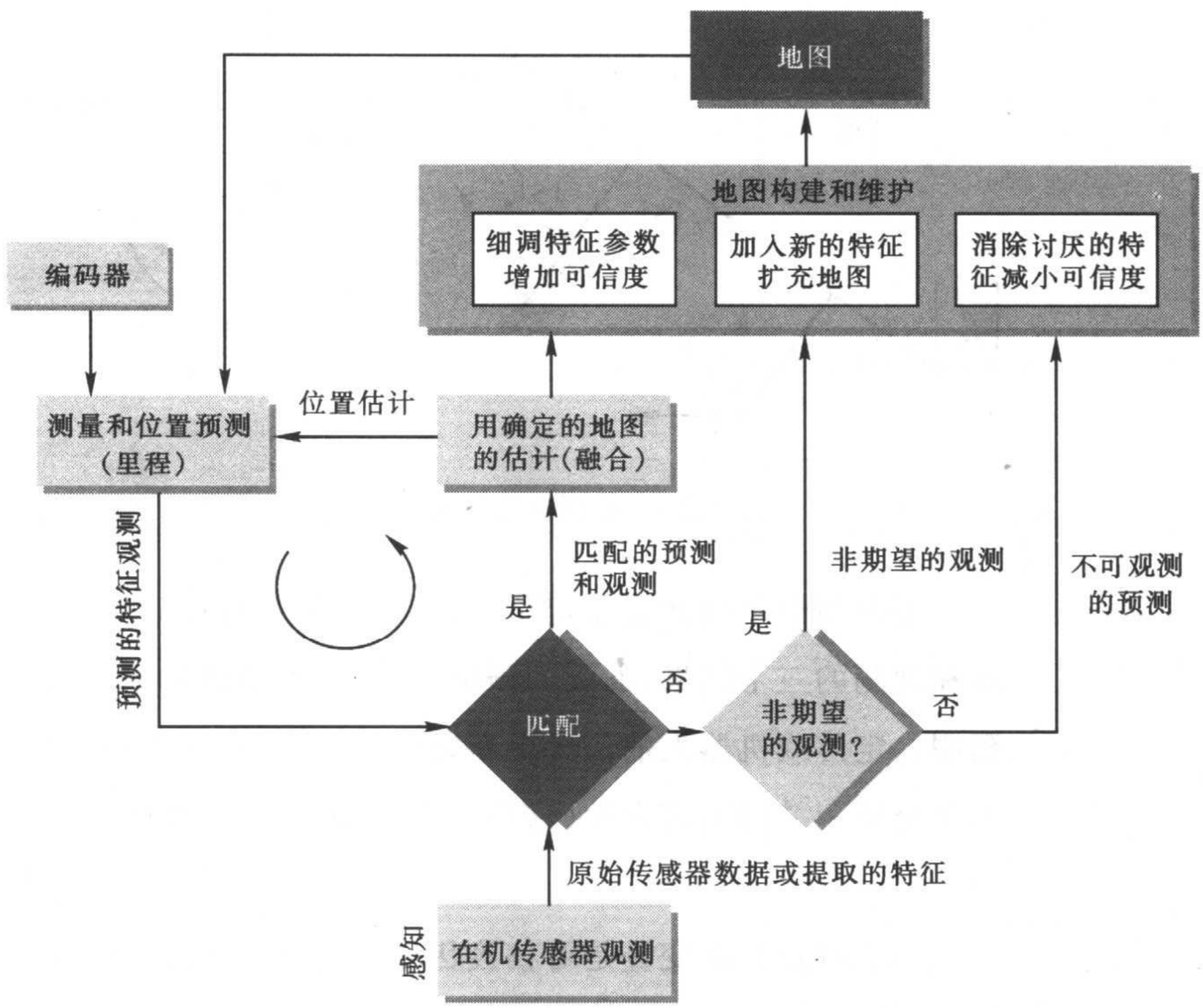


图 5.38 同时定位和地图构建的通用架构[见 23]

预测会影响特征从地图中消去。如以前所讨论的那样,各特定的预测或观测有一个未知的精确值,故它由分布来表示。整个过程必须考虑这些量的不确定性。

我们创建的新型地图,不仅像以前地图所作的那样地图中有特征,而且还有各特征确是环境一部分的概率可变度。我们用随机特征位置 $\hat{z}_t$ 的集合 n 表示这个新的地图 M,各具协方差 Σ_t 和 0 与 1 之间的关联可信度因子 c_t,后者量化环境中特征存在的信任度(图 5.39):

$$M=\{\hat{z}_t,\Sigma_t,c_t\mid(1\leqslant t\leqslant n)\} \tag{5.69}$$

与以前用于卡尔曼滤波器的地图相反,我们不假定精确地知道地图 M,因它将随时由不确定机器人创建。这就是为什么特征 $\hat{z}_t$ 用相关协方差矩阵 Σ_t 进行描述的原因。

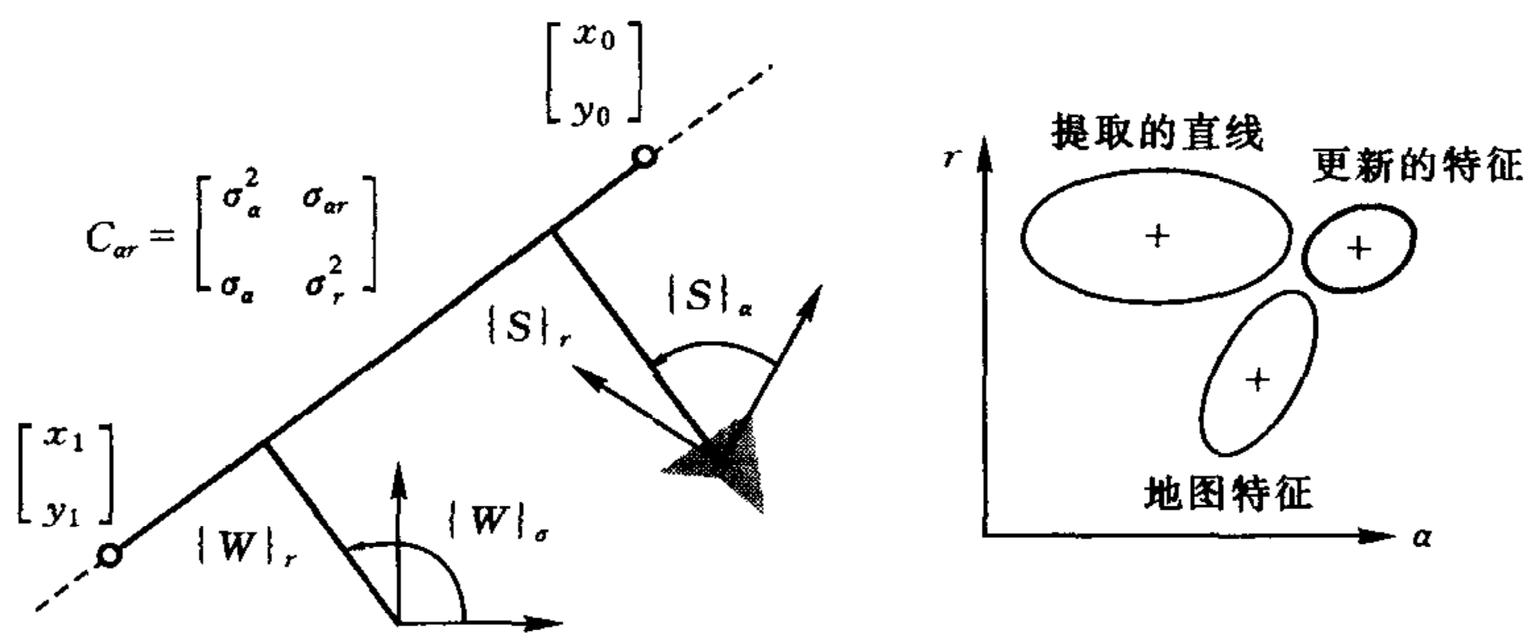


图 5.39 地图中的不确定性

正像与卡尔曼滤波器定位一样，在匹配步骤中，得到有关测量预测和观测的三个结果：**匹配的预测和观测、非期望的观测和不可观的预测**。定位或机器人的位置更新，进行如前。然而，利用所有三个结果和全部相关的不确定性（详见参考文献[23]），现在也同样更新地图。

一个有意义的变量是可信度因子 c_t，它支配被画的特征确在环境中的似然度。为什么机器人在观测到的特征与一个特殊地图特征的匹配中失败会减少地图特征的可信度呢？同样，为什么机器人在匹配被画特征中的成功会增加被画特征是“正确”的机会呢？在参考文献[13]中，建议用以下函数计算可信度：

$$c_t(k) = 1 - e^{-\left(\frac{n_s}{a} - \frac{n_u}{b}\right)} \tag{5.70}$$

式中 a 和 b 定义了学习和遗忘率，n_s 和 n_u 是分别是到时刻 k，被匹配和不可观的预测的次数。协方差矩阵 Σ_t 的更新可以和前节所见的位置更新一样进行。在地图构建中，特征位置和机器人位置是强相关的，这迫使我们使用**随机地图**，在该地图中，必须在各周期更新所有的互相关性[55,113,136]。

随机地图由堆叠系统的状态向量

$$X = [x_r(k) x_1(k) x_2(k) \cdots x_n(k)]^T \tag{5.71}$$

和系统状态协方差矩阵

$$\Sigma=\begin{bmatrix} C_{rr} & C_{r1} & C_{r2} & \cdots & C_{rn} \\ C_{1r} & C_{11} & \cdots & \cdots & C_{1n} \\ C_{2r} & \cdots & \cdots & \cdots & C_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ C_{nr} & C_{n1} & C_{n2} & \cdots & C_{nn} \end{bmatrix} \tag{5.72}$$

组成。式中索引 r 代表机器人，$i=1,\cdots,n$，代表地图中特征。

与基于一个先验准确地图的定位相反，在随机地图的情况下，单个机器人进行自动构建地图时必须维护和更新互相关。在各定位周期，也必须更新机器人-特征和特征-机器人的互相关。简而言之，这个最优方法，要求地图中每一个值依赖于每一个其他的值。由此提出了理由，即自动作图问题这种完整的解决方案，甚至超出了今天计算资源可达的范围。

5.8.2 其他的构图技术

移动机器人学的研究团体在自动作图问题上已花了巨大的研究力量，在许多环境中已展示了工作系统，却没有解决前述的完备的随机地图问题。移动机器人学研究领域极其广泛，本书不对此领域提供全面的综述，我们代之以在下面提出与自动作图相关的两个关键性的考虑，并对几个自动作图解决方案，为克服这些困难所采取的途径作简要的讨论。

5.8.2.1 环状的环境

对自动作图要战胜的、单个最困难的挑战可能是正确地画出环状的环境。问题是简单的：给定一个具有一个或多个回路或环路（例如，四个大厅交叉组成一个矩形），建立整个环境全局一致的地图。

这个问题难在于自动作图系统的基本行为：它们建立的地图不是理想的。而且，给定任何局部的不完整性，则这种不完整性会在整个时间里累加起来，可能在宏观级别上产生地图和现实环境之间任意大的**全局**误差，如图 5.40 所示。这种全局误差常常与机器人定位

和导航无关。必竟受扭曲的地图只要局部误差有界，仍会很好地为机器人服务。然而，一个极大的环路最终仍应回到相同的点。机器人在它的地图中，必须能够注意到该事实。所以，全局误差在环状情况下确实至关紧要。

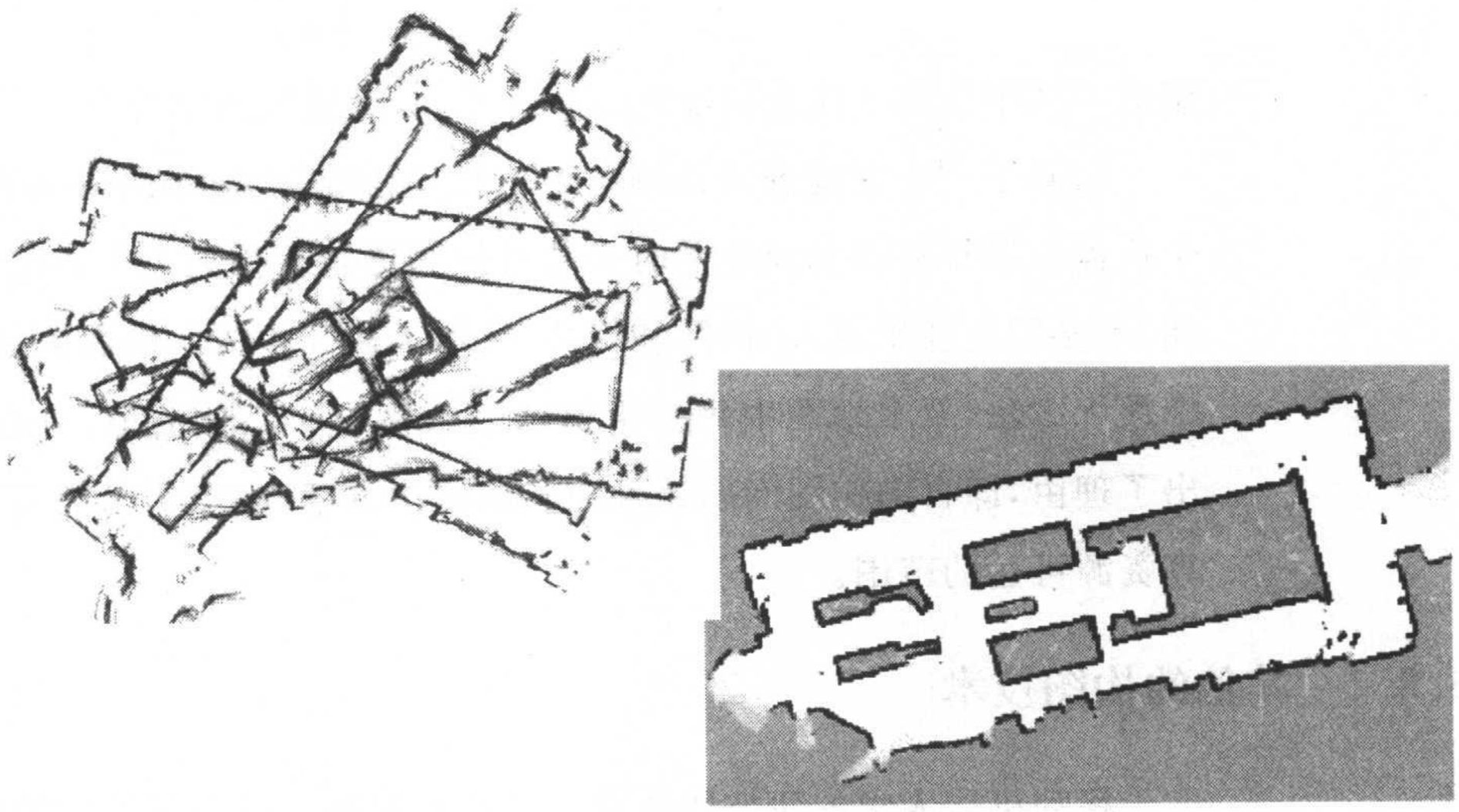


图 5.40 环状环境：具有小的局部误差的不成熟的局部作图策略产生有重大误差的全局地图，如左边真实环境运行所示。应用拓扑校正，提取了在右边的栅格图（图片由 S. Thrum[142] 提供）

在某些最早企图解决环状环境问题的著作中，Kuipers 和 Byun[94] 使用过纯拓扑的环境表示方法，声称拓扑表示只获取最抽象、最重要的特征，避免了大量无关紧要的细节。当机器人到达可能是以前访问过和画过的拓扑节点（例如相同的特征）时，机器人就假设它确已回到了同一节点。为了检测这个假设，机器人明确地规划并移动到相邻节点，看它的感知读数是否与循环假设一致。

随着近来计量地图的普及，诸如固定分解的栅格表示，循环检测策略不是直截了当的。在宣称解决了循环检测问题的绝大多数的自主作图系统中，发现了两个重要特征。第一，如许多新近的系统一样，这些机器人趋向于累计最近的感知经历，建立小规模的局部子

图[51,74,154]，将各子图在机器人位置更新期间处理成一个单独传感器。这个方法的优点有两方面。因为里程表在短距离比较准确，特征的相关登记和原始传感器在子图中读数会是十分准确。此外，机器人将会有一个建好的虚拟的传感器系统，它具有比实际传感器系统的测距大很多的范围。在一定意义上，这个策略通过机器人能很好处理的增加地图规模的办法，至少缓解了很大的环状环境问题。

处理环状环境的第二个新近技术，事实上是回到拓扑表示方法。某些新近的自动作图系统力图利用拓扑与计量子图集合进行关联办法，来辨识环形，即首先在拓扑级别上明确地辨识环路。在参考文献[51]的情况下，例如，由在已知的路标位置按下按钮的人，来识别拓扑级的环路。在参考文献[74]的情况下，拓扑级环路通过子图间履行对应性测试来决定。假设当对应性很好时，两个子图表示了同一地方。

基于已知的拓扑方法，人们肯定可以想象其他的增强方法。例如，可以用在 5.7 节描述的全局唯一定位方法辨识拓扑的正确性。值得注意的是，目前自动作图的研究在许多方法中已回到了基本的拓扑正确性问题。这在十多年前，就是移动机器人学某些最早的自动作图研究的核心。当然，与早期工作不同，今天的自动作图结果与环境高保真度的几何地图相结合，以正确的环形检测而自豪。

5.8.2.2 动态环境

另一个挑战不仅仅扩展到现存的自主的作图解决方案，而且扩展到随机地图的基本形成方法。所有这些策略，倾向于假定环境或者是不变的，或者实际上变化不显著。相对于某些环境，这种假设确实是有效的。例如，在早晨 3 点钟，大学的计算机科学系。然而在许多情况下，该假设是不正确的。在人类普遍聚集的场所，即广为开放的空间情况下，存在自由空间快速变化，而且大多数传感器读数代表了对瞬态人群的检测，而不是对固定表面（如边界墙）的检测。另外一类的动态环境，诸如工厂车间和仓库那样的空间，在那里，当货物进出移动时，所存储的物体要按天重新定义路径的拓扑。

在所有这种动态环境中，自动的作图系统应该捕获由它的传感器检测到的易见物体。而且，机器人应该具有灵活性，当这些易见的物体位置改变时，修改它的地图。**连续作图**或动态环境作图的主题，是在某种程度上对不熟悉的环境自动作图的成功策略的一个直接副产物。例如，在用可信度因子 c_t 机制的随机作图情况下，在外表上完成初始的地图创建过程后，可信度方程可以连续提供有关各种被画特征的存在性的概率反馈。因此，通过简单的连续操作，作图系统可以变成一个地图修改系统。当然，如果作图系统是实时和增量式的，这是最有效的。如果地图构造要求离线全局优化，那么，对地图进行小粒度、增量式调整的愿望是比较难满足的。

前面我们说过，作图系统应该只捕捉由它的传感器检测到的显而易见的物体。比如，在环境中处理人类检测的一个常见的论据是：像 c_t 这样的机制可以照顾到一开始不值得作图的全部特征。例如，在参考文献[157]中，作者基于局部占用栅格(称**证据栅格**)和全局占用栅格的集合开发了一个系统。每次用机器人最新近的局部栅格更新全局占用栅格的区域。如果在那些相同的位置局部占用栅格没有检测到物体(以高的可信度)，那么在全局占用栅格释放无关系的被占用的单元。

然而，检测易见特征问题的一般解，通常求助于感知问题的解。当机器人的传感器能可靠地检测墙和人之间的差别时，例如，用一个视觉系统，那么在动态环境中作图问题就会意味深长地变得更直接明了。

我们仅讨论了自动作图的两个重要考虑。还有大量针对一般性地图构建和定位问题的研究活动[22,23,55,63,80,134,147,156]。但是，只有少数几个组工作在概率地图构建(即，随机地图)的一般问题上。至今，还没有找到一致且绝对的一般解答。在以后几年，这个领域肯定会产生有重大意义的新成果，并且随着机器人感知功能的改善，我们预料，该领域的成果一定是最大的。

第 6 章　规划与导航

6.1　引言

本书已将精力集中于对鲁棒性、机动性至关重要的移动机器人的重要因素：运动的运动学；确定机器人环境背景的传感器；相对于机器人地图的定位技术。现在，我们把注意力转向机器人的认知。一般来说，认知表示系统利用有目的的决策和执行，实现最高级别的目标。

在移动机器人的情况下，与鲁棒的机动性直接相关的认知特性是**导航能力**。给定关于环境的部分知识和一个目标位置或位置序列，导航包括机器人的动作能力，它根据机器人的知识和传感器值使之尽可能有效和可靠地到达它的目标位置。本章的重点是如何能结合前几章的工具，解决这个导航问题。

在移动机器人学的研究团体内，已经提出了非常多的解决导航问题的方法。当我们从该研究背景取样时，我们会明白，事实上所有这些方法之间存在高度的相似性，尽管它们在表面上显得十分不同。在各种不同的导航体系结构中，关键的差别在于它们将问题分解为较小子单元的方式。在下面的 6.3 节中，我们描述最流行的导航体系结构，并对比它们相对的优缺点。

然而，在 6.2 节中，我们将首先讨论移动机器人导航所要求的两

个关键的辅助能力。给定一张地图和一个目标位置，**路径规划**(path planning)涉及辨识轨迹。在执行时，该轨迹促成机器人到达目标位置。因为机器人必须决定为完成长期目标该做什么，所以路径规划是一种战略性的问题求解能力。

第二个能力是同样重要的，但它处在相对的战术的一端。给定实时的传感器读数，避障意味着为避免碰撞调整机器人的轨迹。种类繁多的方法已展示了它们能胜任避障，我们将概述其中几个方法。

6.2 导航能力：规划和反应

在人工智能团体中，规划与反应常被视为是相对的，甚至是相反的两种方法。事实上，当被应用于实际系统，如移动机器人时，规划与反应有着非常强的互补性，各自对另一方的成功都极其重要。机器人的导航难题涉及到为达到它目标位置，执行动作(或规划)的方针。在执行期间，机器人必须对一些不可预见的事件(如障碍)以某种方式做出反应，使之仍然达到目标。如果没有反应，规划的结果将是徒劳的，因为机器人在物理上将永远不会达到目标。如果没有规划，反应的结果不能指导整个机器人的行为以达到它长远的目标——机器人又一次达不到它的目标。

导航问题的信息论方程使这个互补性清晰易懂。假设机器人 M 在 i 时刻有一张地图 M_i 和一个初始的信任度状态 b_i。机器人的目标是到达位置点 p，同时应满足某些时间约束：$loc_g(R)=p$；($g\leqslant n$)。因此机器人必须在时间步 n 或之前到达目标位置 p。

尽管机器人要到达的目标在物理上是清楚的，但机器人只真实感知其信任度状态而不是它的物理位置。因此我们将到达位置 p 的目标映射成达到的信任度 b_g，相当于 $loc_g(R)=p$ 的信任度。利用这个方程，规划 q 无非是从 b_i 到 b_g 的一条或多条轨迹。换句话说，

如果规划 q 的执行是从与 b_i 和 M_i 两者一致的环境状态开始，则该规划 q 会促使机器人信任度状态从 b_i 到 b_g 的转变。

当然，问题是后者的条件可能不满足。机器人的位置与 b_i 可能不十分相符，甚至更可能的是 M_i 不完备或者不正确。而且，现实环境是动态的，即使 M_i 作为时间上单个瞬象是正确的，有关 M 如何随时间变化的规划器模型却常常是不理想的。

虽然如此，为实现目标，在规划执行期间机器人必须纳入所获得的新的信息。随着时间的推进，环境不断发生变化，机器人传感器聚集新的信息。这正是反应变为有重要作用的地方。在最好的情况下，为了校正规划好的轨迹，反应会局部地调整机器人的行为使机器人依然到达目标。有时，一些不可预测的新的信息会要求机器人改变战略规划。所以，当收到新的信息时，理想情况下规划器也纳入这些新的信息。

最好的情况是规划器会实时地容纳每一片新的信息，并即时产生适当地反映新信息的新规划。这是理论上的极端情况。至此，规划概念和反应概念合并，并称为**集成的规划与执行**。我们将在6.3.4.3节讨论这个议题。

贯穿机器人结构的讨论中，有一个有用的概念。这个概念涉及到不论何时当存在一个解决方案时，特殊的设计决策是否牺牲了系统到达期望目标的能力。这个概念被称为**完备性**(completeness)。更正式地说，一个机器人系统是**完备**的，当且仅当对于所有可能的情况(例如，初始信任度状态、地图和目标)，若存在一条到达目标信任度状态的轨迹，则系统会到达目标信任度状态(更详细的内容见参考文献[27])。所以，当系统是非完备时，那么至少存在一个实例，对此，尽管存在一个解决方案，但系统无法产生该方案。如大家所期待的，实现系统的完备性是一个富有挑战性的目标。在表示与推理的层次上，因计算复杂性，常常牺牲完备性。在分析上，重要的是要了解各特定系统如何兼顾完备性。

在下面的各节中，我们描述当规划与反应用于移动机器人路径规划和避障时关键的几个方面，且阐述具有代表性的决策如何影响整个系统潜在的完备性。有关这方面更详细的介绍，请参看[21,30，第 25 章]。

6.2.1 路径规划

甚至在可供应的移动机器人出现之前，由于它在工业机械手领域中的应用，已经大量地研究了路径规划。令人感兴趣的是，拥有 6 个自由度的机械手的路径规划问题远比在平坦环境中运作的差动机器人复杂。所以，尽管我们可以从为机械手所创造的技术中获得启发，但移动机器人所用的路径规划算法，由于大幅度降低了自由度，趋向于较简单的近似。然而，由于工厂生产线上高生产力的经济影响，工业机器人通常以最快可能的速度运行。所以，动力学和它们的运动学都是非常重要的，这进一步使路径规划与执行复杂化。相反，许多移动机器人在很低的速度下运行，以至在路径规划时极少考虑动力学，这进一步简化了移动机器人瞬态问题。

配置空间 确实，机械手机器人和大多数移动机器人的路径规划，形式上是在称之为**配置空间**的表示方法中完成的。假设机器人手臂(如 SCARA 机器人)有 k 个自由度，则机器人的每个状态或配置，可用 k 个实值 $q_q,\cdots,q_k$ 予以描述。这 k 个值可看作是在称为机器人配置空间 C 的 k 维空间中的一个点 p。这种描述是方便的，因为它允许我们用单一的 k 维点，描述复杂的三维机器人形状。

现在考虑一个在环境中运动的机器人手臂。环境中的工作空间(即它的物理空间)包含已知的障碍物。路径规划的目标是在物理空间中找到一条从初始点位置到终点位置的路径，避免与所有障碍物碰撞。这是一个在物理空间中是很难想象和解决的问题，尤其当维数 k 增大时。但在配置空间中，问题是直截了当的。如果将**配置空间障碍物** O 定义为 C 的子集，在 C 中机器人手臂撞击某物体，我们

可计算机器人可安全移动的自由空间为:$F=C-O$。

图 6.1 表示了双杆平面机器人的物理空间和配置空间。机器人的目标是从起始位置到终点位置,移动其末端执行器。图示的配置空间是二维的,因为各关节可以是 0～π 的任何位置。容易看出,在 C 空间中的解,是一条始终保持在机器人手臂自由空间内,从起始点到终点的直线。

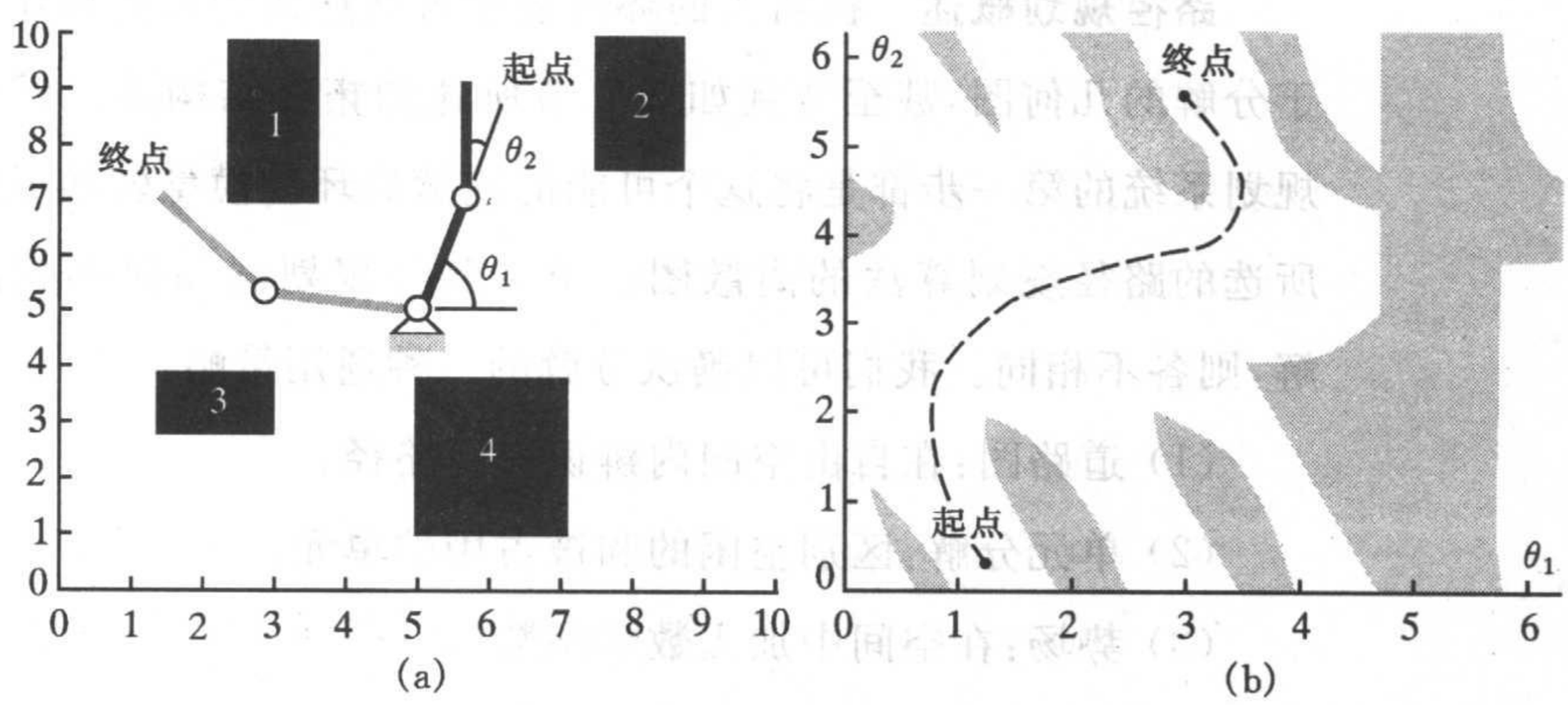


图 6.1 物理空间(a)和配置空间(b):(a)双杆平面机器人必须从配置的起始点移动到终点,所以运动过程受障碍物 1 到 4 的约束;(b)相应的配置空间以关节坐标(角 θ_1 和 θ_2)和一条到达目标的路径表示了自由空间

对于在平地上运行的移动机器人,如第 3 章所述,我们一般用 3 个变量表示机器人位置(x,y,θ)。但正如我们所见,大多机器人使用差动驱动或 Ackerman 操纵系统是非完全的。对这种机器人,非完全约束限制了在各配置(x,y,θ)中机器人的速度($\dot{x},\dot{y},\dot{\theta}$)。关于构造合适的**自由空间**以解决这类路径规划问题,详见参考文献[21, p. 405]。

在移动机器人学中,为路径规划起见,最常用的方法是假设机器人实际上是完全的,这极大地简化了过程。这对差动驱动机器人,更为普通,因为它们可以在适当位置转动。所以,如果机器人的转动位置不是关键的,则可容易地模仿完全的路径。

移动机器人专家通常会在进一步的假设下:即机器人只不过是

一个点，进行规划。因此，我们可以进一步将移动机器人路径规划的配置空间简约为只有 x 和 y 轴的 2D 表示。所有这些简化的结果是：配置空间看起基本上与 2D(即平面)型式的物理空间一样。但有一个重要区别，因为我们已将机器人简化为一个点，所以我们必须将各个障碍物按机器人的半径尺寸进行膨胀。心中有了这个新的、简化的配置空间，我们现在就可以介绍移动机器人路径规划的公用技术。

路径规划概述 机器人的环境表示可从连续的几何描述变到基于分解的几何图，甚至变到如 5.5 节所述的拓扑结构图。任何路径规划系统的第一步都是将这个可能的连续的环境模型转换成适合于所选的路径规划算法的离散图。至于路径规划器如何影响离散分解，则各不相同。我们可以确认分解的三种通用策略：

(1) 道路图：在自由空间内辨识一组路径。

(2) 单元分解：区别空闲的和被占用的单元。

(3) 势场：在空间中加入数学函数。

下面各节提出道路图和单元分解路径规划技术的一些常用实例。注意在各种情况下，完备性是否因特殊的表示方法而牺牲。

6.2.1.1 道路图路径规划

道路图方法是在被称为**路线图**的一维曲线或直线的网络中，获取机器人自由空间的连接性。一旦构造了路线图，它就被用作机器人运动规划的路(路径)段网络。因此，路径规划被缩简为把机器人的初始和目标位置连接到路网，然后搜索从初始位置到目标位置的一系列路径。

路线图方法是特别根据障碍物的几何形状对机器人配置空间进行分解。其困难是要构建一组路径，它们合在一起能使机器人在它的自由空间中行走到任何地方，同时使总路径的数目最小。一般而言，只要以适当的保真度获取到机器人的真正自由度，则在这种分解中就保住了完备性。下面我们描述两种路线图方法，它们以极不相同的路径类型实现了该结果。在**可视性图**的情况下，路径尽可能地

靠近障碍物,且最终路径是极小长度解。在**沃罗诺伊**(Voronoi)图的情况下,路径尽可能地远离障碍物。

可视性图 多边形配置空间的可视性图,是由连接彼此可见的全部顶点对的连线组成(包括像顶点一样的初始和目标位置)。显然,连接这些无阻挡的顶点是它们之间的最短距离。因此,路径规划器的任务就是沿着由可视性图定义(图 6.2)的路径,寻找从初始位置到目标位置的最短路径。

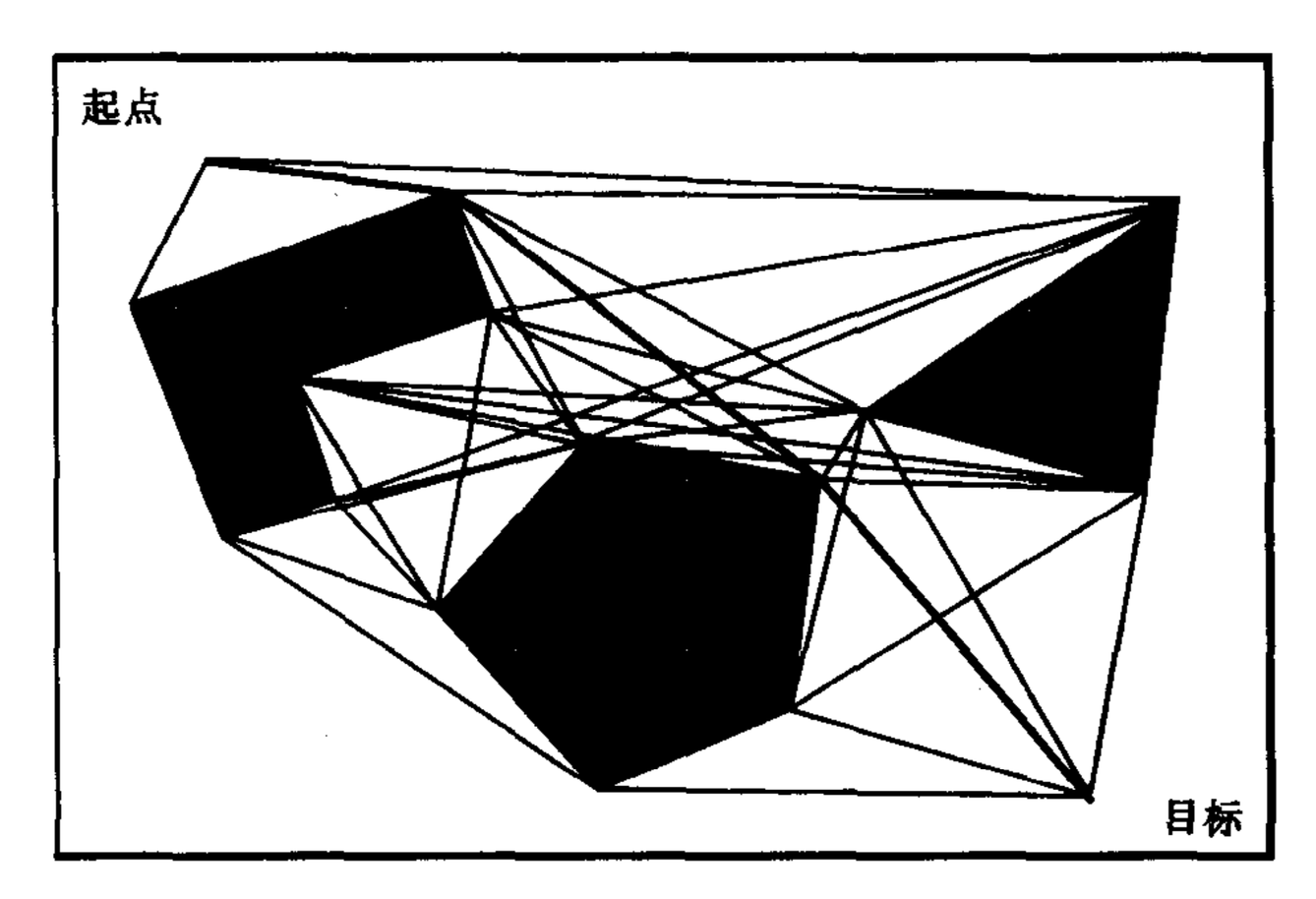


图 6.2 可视性图[21] 图中节点是初始点和目标点及配置空间中障碍物(多边形)的顶点。相互可见的所有节点用直线段连接,从而定义了路线图。这意味着这里也存在沿各多边形直线段的边缘

可视性图在移动机器人学的路径规划中比较普遍,部分是因为实现比较简单。特别在连续的或离散的空间中,当环境的表示把环境中的物体描述成多边形时,可视性图的搜索可容易地使用障碍物的多边形描述。

但是,在使用可视性图进行搜索时,有两个重要的警告。第一,表示的规模和节点以及边缘的数目,会随障碍物多边形的数目而增长。所以,该方法在稀疏环境中极其迅速和有效。但是,当用在密集的居住环境时,与其他技术相比,该方法既慢又无效。

第二个警告是更为严重的潜在的缺陷:由可视性图规划所找到

的答解路径倾向于在走向目标的途中，使机器人尽可能地接近障碍物。更正式地说，我们可以证明可视性图规划按解答路径的长短是最优的。这个强有力的结果也意味着，如按照机器人与障碍物保持合理的距离，则安全保障会因这个最优性而丧失。常用的解决方法是：把障碍物增大到比机器人半径大得多；或者换一种方法，在完成路径规划后，如可能的话，修改解答路径距障碍物的距离。当然，这样作法牺牲了可视性图路径规划的最优长度。

沃罗诺伊图 与可视性图相比，沃罗诺伊图是一种全路线图的方法，它倾向于使图中机器人与障碍物之间的距离最大化。对自由空间中的各点，计算它到最近障碍物的距离。在图 6.3 中，把距离画成离页面的高度（图 6.3*）。当离开障碍物而移动时，高度值增加。

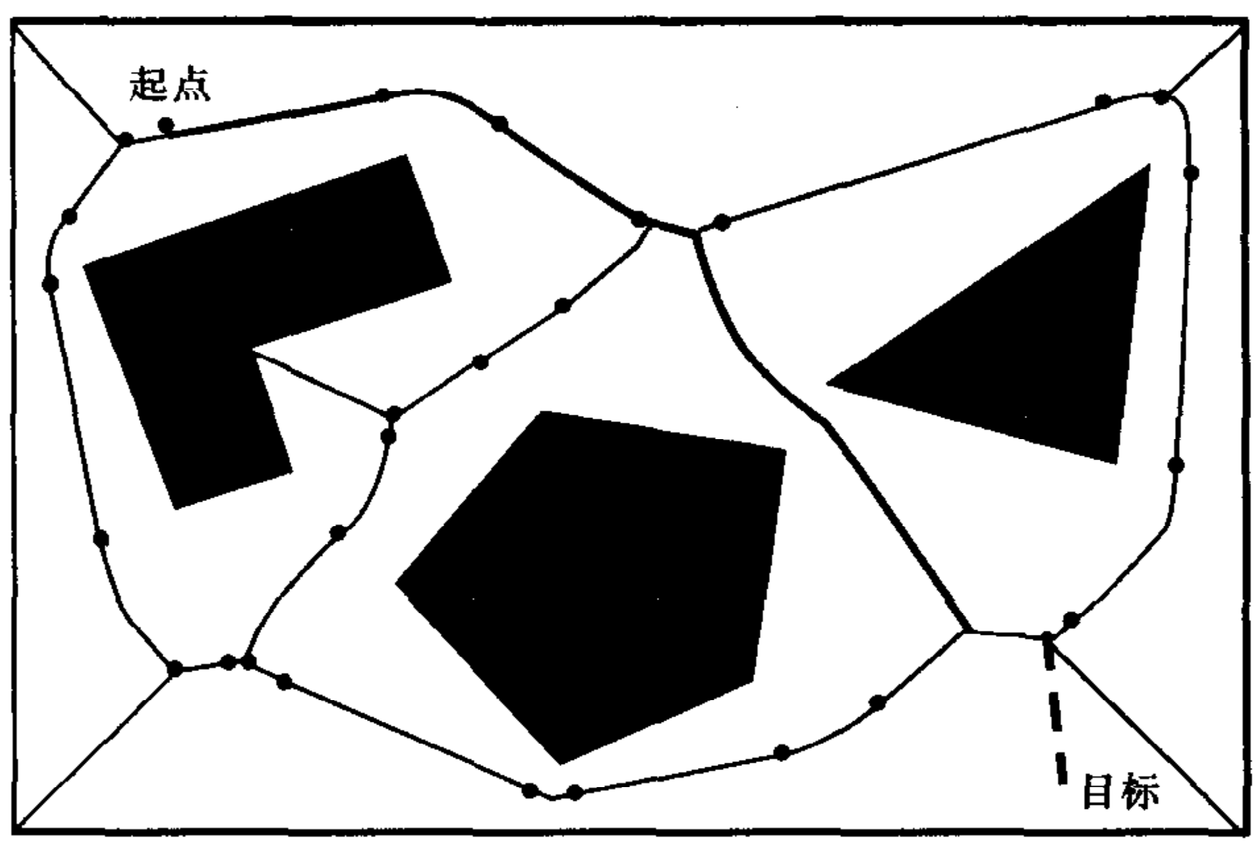


图 6.3 沃罗诺伊图[21]。沃罗诺伊图由直线组成，而直线由距两个障碍物或多个障碍物等距离的所有点构成。初始点 q_{init} 和目标点 q_{goal} 的配置被映射到沃罗诺伊图的 q'_{init} 和 q'_{goal}，各由画直线得到，沿着该直线到障碍物边界的距离增加最快。在沃罗诺伊图上也可选择运动方向，使之距边界的距离增加最快。沃罗诺伊图上的点表示从直线段（两直线间的最小距离）到抛物线段（直线和点之间的最小距离）的过渡

在离两个或多个障碍物等距离的点上，这种距离图就有陡的山脊。

* 图 6.3 是投影图——译者注。

沃罗诺伊图就是由这些陡的山脊点所形成的边缘组成。当配置空间障碍物都是多边形时,沃罗诺伊图由直线和抛物线段组成。在沃罗诺伊路径图上寻找路径的算法,正像可视性图方法一样,是完备的,因为自由空间中路径的存在意味着在沃罗诺伊图上也存在一条路径(即两种方法都确保完备性)。但是,在总长度的意义上,沃罗诺伊图常常远非最优。

在有限距离的定位传感器情况下,沃罗诺伊图有一个重要的弱点。因为该路径规划算法使环境中机器人与物体之间的距离最大化,机器人上的任何短距传感器会有感知不到周围环境的危险。如果这种短距传感器用于定位,则从定位的观点看,所选的路径会很差。另一方面,我们可以设计可视性图,使机器人尽可能像我们所期望的那样靠近地图中的物体。

但沃罗诺伊图有一个重要的难得优点,即沃罗诺伊图方法具有超越其他大多数避障的技术:**可执行性**。通过沃罗诺伊图规划,给定一个特殊的已规划路径,配备有距离传感器(如激光测距仪或超声波传感器)的机器人可以使用简单的控制规则,跟踪在物理世界中沃罗诺伊的边缘。这些规则与创建沃罗诺伊图的规则相匹配:即机器人使其传感器值的局部极小值最大化。这种控制系统会自然地使机器人保持在沃罗诺伊边缘上,所以基于沃罗诺伊的运动减少了编码器的不准确性。沃罗诺伊图的这种有趣的物理性质,即通过在未知的沃罗诺伊边缘上寻找和移动,已经被用于引导环境的自动作图,然后构建与环境一致的沃罗诺伊图[59]。

6.2.1.2 单元分解路径规划

支持单元分解的思想是区分几何区或单元之间的差别,即把单元分为自由的和被物体占用的区间。基本的单元分解路径规划算法可概括如下[30]:

- 将 F(图)分为简单的、称之为“单元”的连接区。
- 确定哪些开放单元是相邻的,并构建一个“连接图”。

- 找出初始和目标配置所在的单元，并在连接图中搜索一条连接初始和目标单元的路径。
- 用合适的搜索算法，从找到的单元序列中计算各单元内的路径，例如，穿过单元边界的中点的距离或沿直线随墙的运动或移动的序列。

单元分解方法的一个重要方面是单元之间边界的安置。如果边界安置成是环境结构的函数，使得分解是无损的，则该方法称为**精确的单元分解**。如果分解形成实际地图的近似，则系统称为**近似的单元分解**。在5.5.2节中，当它们应用于设计定位的地图表示方法时，我们描述了这些分解策略。这里，我们再次简单地概述这两种分解技术，并更详细地提出相对于路径规划的有关优缺点。

精确的单元分解 图6.4描述了精确的单元分解。这里，单元

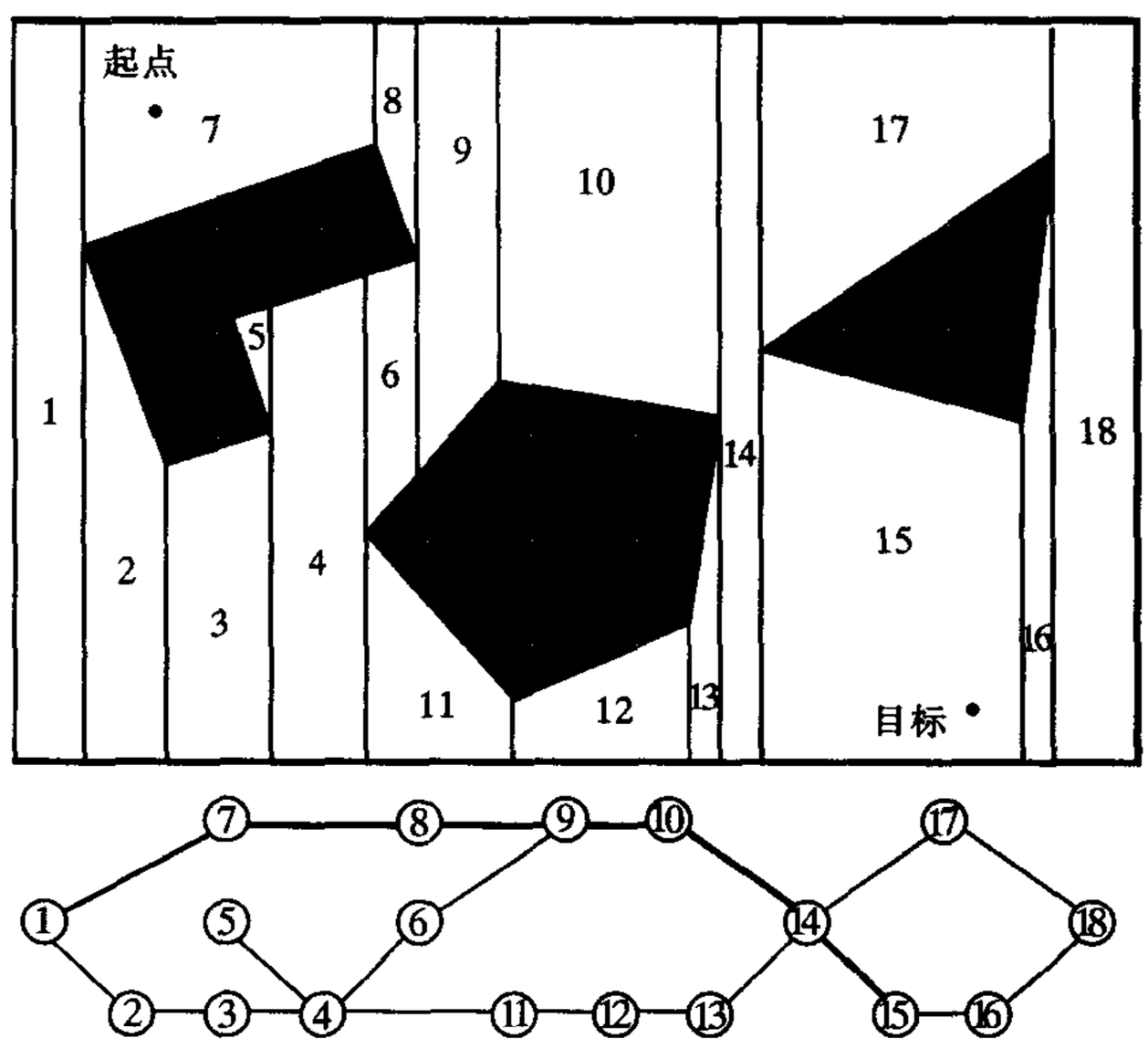


图6.4 精确单元分解举例

的边界建立在几何临界性的基础上。最后得到的单元或各是完全自由的，或各被完全占用。所以，在网络中路径规划与上述基于路线图

的方法一样是完备的。支持这种分解的基本抽象概念是:在自由空间的各单元内,机器人的特殊位置无关紧要,重要的是机器人从各自由空间单元走向其相邻自由单元的能力。

精确单元分解的主要缺点是单元的数目。正如基于道路图的系统一样,整个路径规划的计算效率取决于环境中物体的密度和复杂性。主要的优点是这个同一相关性的结果。在极稀疏的环境中,即使环境的几何尺寸很大,单元的数目仍然少。因此在大的稀疏环境中,这种表示方法将是有效的。实事求是地说,由于实现的复杂性,精确单元分解技术在移动机器人中应用较少,尽管在高度希望无损表示时它保留有可靠选择的余地。例如,保持充分的完备性。

近似单元分解 相比较而言,近似单元分解是移动机器人路径规划中最普遍的技术之一。这部分地是由于基于栅格的环境表示的普遍性。这些基于栅格的表示方法本身是栅格尺寸固定的分解,所以它们与环境的近似单元分解相同。

这种方法中最流行的形式是固定尺寸的单元分解,如第 5 章图 5.15 所示。单元的尺寸不依赖于环境中的特殊物体。因此,由于棋盘格不精确的性质,狭窄的通路可能被丢失。事实上,由于所用单元尺寸很小(如各边 5 cm),这是个罕见的问题。固定尺寸单元的分解最大的好处是路径规划的计算复杂性低。

例如,NF1(Navigation Function1 导航函数 1——译者注),常被称为**野火**(grassfire),它在这种固定尺寸的单元阵列中寻找路径,是一种有效且易于实现的技术[96]。算法简单地使用从目标位置向外的前波扩展,对各单元标记到目标位置的距离[79]。这个过程连续地进行,直至到达相应于机器人初始位置的单元。在这一点上,路径规划器可估计机器人到目标位置的距离,而且通过简单地将相邻的、并更接近目标的单元连接在一起,就可得到一条特定解的轨迹。

假设整个单元阵列保存在存储器中,当从初始位置开始搜索到目标位置的最短离散路径时,各个单元只被访问一次。所以搜索只

是单元数目的线性关系。因此复杂性既不依赖于环境的疏密，也不依赖于环境中物体形状的复杂性。形式上，该野火变换是简单的宽度优先搜索，它是在邻接阵列受约束的环境中实现的。有关宽度优先搜索和其他图形搜索技术更多信息，参考文献[30]。

固定分解的基本代价是存储器。对于一个大的空间来说，即使是稀疏的，这个栅格也必须全面地表示出来。实际上，由于计算机内存降价，近年来这种缺陷有所缓和。Cye 机器人是商用机器人的一个例子，它采用了躲避已知障碍物，且以已知路径为优先[42]的复杂野火算法。全部的路径规划都是在 2D 2 cm 固定单元分解的环境上进行。

第 5 章的图 5.16 举例说明了一种尺寸可变的近似单元分解方法。自由空间外部由矩形限定，内部由 3 个多边形限定。矩形被递归地分解为更小的矩形，每次分解产生 4 个相同的新矩形。在各分辨率的级别上，只有那些内部完全处在自由空间的单元才被用于构建连接图。在这种自适应表示方法下，路径规划可按递阶方式进行。以粗分辨率开始，逐步提高*分辨率，直至路径规划器确认了一个解答，或者获得了一个限定的分辨率（例如 $k\times$ 机器人尺寸）。与精确的单元分解法相比，近似的方法可能牺牲完备性，但它很少涉及数学，所以比较容易实现。与固定尺寸单元分解相比，变尺寸单元分解适合于复杂的环境。因此，稀疏环境将包含适当的少量单元，耗费极少的存储器。

6.2.1.3 势场路径规划

势场路径规划跨越机器人的地图建立一个场或梯度，该地图引导机器人从多个先验位置到达目标位置[21]。这种方法最初是为机器人机械手的路径规划而创建的，并常用于多个变量下的移动机器人领域。势场法把机器人处理成在人工势场 $U(q)$ 影响下的一个点。

* 原文为减少，有误——译者注。

像球滚下山一样,机器人跟随着场移动。目标(空间中的极小值)表现为对机器人的引力,障碍物表现为峰值或斥力。所有这些力的叠加,施加于机器人。在大多数情况下,机器人被假定为配置空间中的一个点(图 6.5)。这样一种人工势场平滑地引导机器人趋向目标,同时避免碰撞已知的障碍物。

重要的是要注意,这不只是路径规划。最终产生的场也是机器人的控制律。假定机器人能够相对于地图和势场定位自身的位置,那么根据势场,它也可经常地确定其下一个所要求的动作。

支持所有势场法的基本思想是:机器人被吸引向目标,同时被先前已知的障碍物所排斥。如果在机器人移动中出现新的障碍物,为了集成这个新的信息,应更新势场。在最简单的情况下,我们假设机器人是一个点,因此忽略机器人的方向 θ,最后所得的势场只是二维的(x,y)。我们假定一个可微的势场函数 $U(q)$,则可得到作用于位置 $q=(x,y)$的相关人工力 $F(q)$:

$$F(q)=-\nabla U(q) \tag{6.1}$$

这里$\nabla U(q)$表示在位置点 q 处 U 的梯度向量。

$$\nabla U=\begin{bmatrix}\dfrac{\partial U}{\partial x}\\ \dfrac{\partial U}{\partial y}\end{bmatrix} \tag{6.2}$$

作用在机器人的势场,计算成是目标的引力场和障碍物的斥力场之和:

$$U(q)=U_{\text{att}}(q)+U_{\text{rep}}(q) \tag{6.3}$$

相似地,力也可被分离成吸引和排斥两部分:

$$F(q)=F_{\text{att}}(q)-F_{\text{rep}}(q)=-\nabla U_{\text{att}}(q)+\nabla U_{\text{rep}}(q) \tag{6.4}$$

吸引势位 例如,一个吸引势位可以定义为抛物线函数:

$$U_{\text{att}}(q)=\frac{1}{2}k_{\text{att}}\cdot\rho_{\text{goal}}^{2}(q) \tag{6.5}$$

式中,k_{att}是一个正的比例因子,$\rho_{\text{goal}}(q)$表示欧氏距离 $\|q-q_{\text{goal}}\|$。

吸引势位是可微的，形成吸引力 F_{att}：

$$F_{att}(q)=-\nabla U_{att}(q) \tag{6.6}$$

$$=-k_{att}\cdot\rho_{goal}(q)\nabla\rho_{goal}(q) \tag{6.7}$$

$$=-k_{att}\cdot(q-q_{goal}) \tag{6.8}$$

当机器人到达目标点时，该力线性地收敛至 0。

排斥势位 支持排斥势位的思想是产生离开所有已知障碍物的力。当机器人越靠近物体时，排斥势位应该很强；但当机器人远离物体时，它不应影响机器人的运动。这种斥力场的一个例子是：

$$U_{rep}(q)=\begin{cases}\frac{1}{2}k_{rep}\left(\frac{1}{\rho(q)}-\frac{1}{\rho_0}\right)^2 & \text{如果 } \rho(q)\leqslant\rho_0\\ 0 & \text{如果 } \rho(q)\geqslant\rho_0\end{cases} \tag{6.9}$$

式中 k_{rep} 也是一个比例因子，$\rho(q)$ 是从 q 点到物体的最小距离，ρ_0 是物体的影响距离。排斥势位函数 $U_{rep}(q)$ 是正的或零，当 q 更接近物体时，它趋于无穷大。

如果物体的边界是凸的，且分段可微，$\rho(q)$ 在自由配置空间中处处可微，则产生斥力 F_{rep}：

$$F_{rep}(q)=-\nabla U_{rep}(q) \tag{6.10}$$

$$=\begin{cases}k_{rep}\left(\frac{1}{\rho(q)}-\frac{1}{\rho_0}\right)\frac{1}{\rho^2(q)}\frac{q-q_{obstacle}}{\rho(q)} & \text{如果 } \rho(q)\leqslant\rho_0\\ 0 & \text{如果 } \rho(q)\geqslant\rho_0\end{cases}$$

作用在一个承受引力和斥力的点机器人，其合力为 $F(q)=F_{att}(q)+F_{rep}(q)$，它使机器人离开障碍物趋向目标（图 6.5）。在理想条件下，通过设置一个正比于场力向量的机器人速度向量，与球绕过障碍物并向山下滚动一样，可平滑地引导机器人趋向目标。

但这种方法有很多局限性。根据障碍物形状和大小，会出现局部极小；如果物体是凹的，就会出现另一个问题，它可能产生几个最小距离值 $\rho(q)$ 同时存在的情况，导致离物体最近的两个点来回震荡，明显地牺牲了完备性。势场特性更详细的分析，可参阅文献[21]。

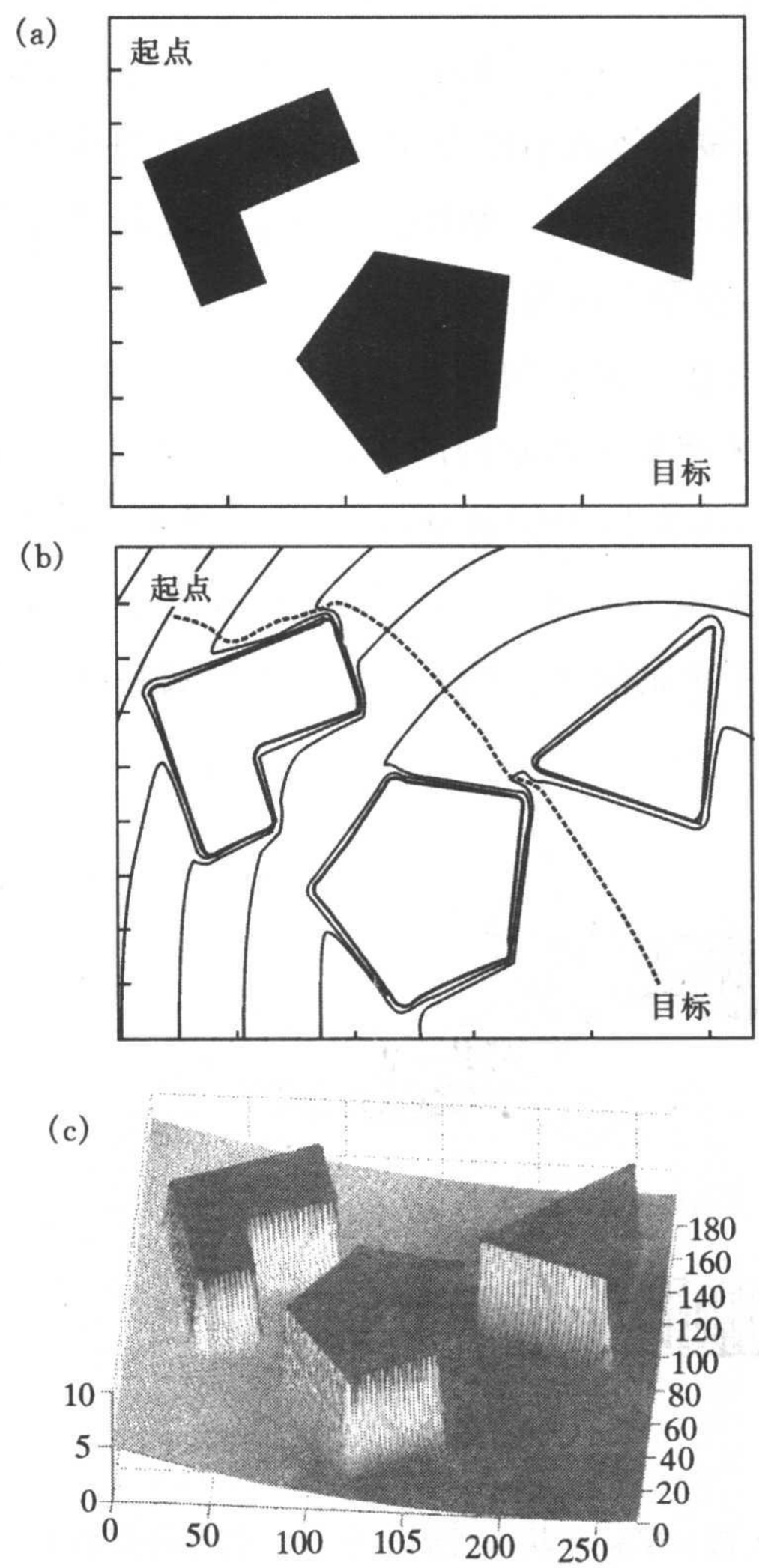


图 6.5 由吸引目标和三个障碍物产生的典型势场[21]。(a)障碍物的配置,起始点(左上)和目标点(右下);(b)等势位图和由势产生的路径;(c)由目标吸引器和障碍物产生的最终势场

扩展的势场法 Khatib 和 Chatila 提出了扩展的势场法[84]。与所有的势场法相似,这个方法利用了来源于人工势场的引力和斥力。但在基本势场上附加了两个场:**转动势场**和**任务势场**。

转动势场假定，斥力是距障碍物距离和机器人相对于障碍物的方向的函数，这是通过一个增益系数实现的。当障碍物与机器人行走方向平行时，该系数减小斥力，因为这样一个物体不会对机器人的轨迹造成一个即时的威胁。结果是增强了沿墙跟踪能力。这对早期势场法的实施，曾是个问题。

任务势场考虑了当前的机器人速度，并由此排除了那些根据近期势能对机器人速度无影响的障碍物。当机器人前方扇形区(称做Z)无障碍物时，再次作比例调整，将所有的障碍物势能加倍。扇形区 Z 可被定义为下一次移动时，机器人快速通过的区域。结果可能是穿过空间更平滑的轨迹。图 6.6 描述了经典势场与扩展势场对比的实例。

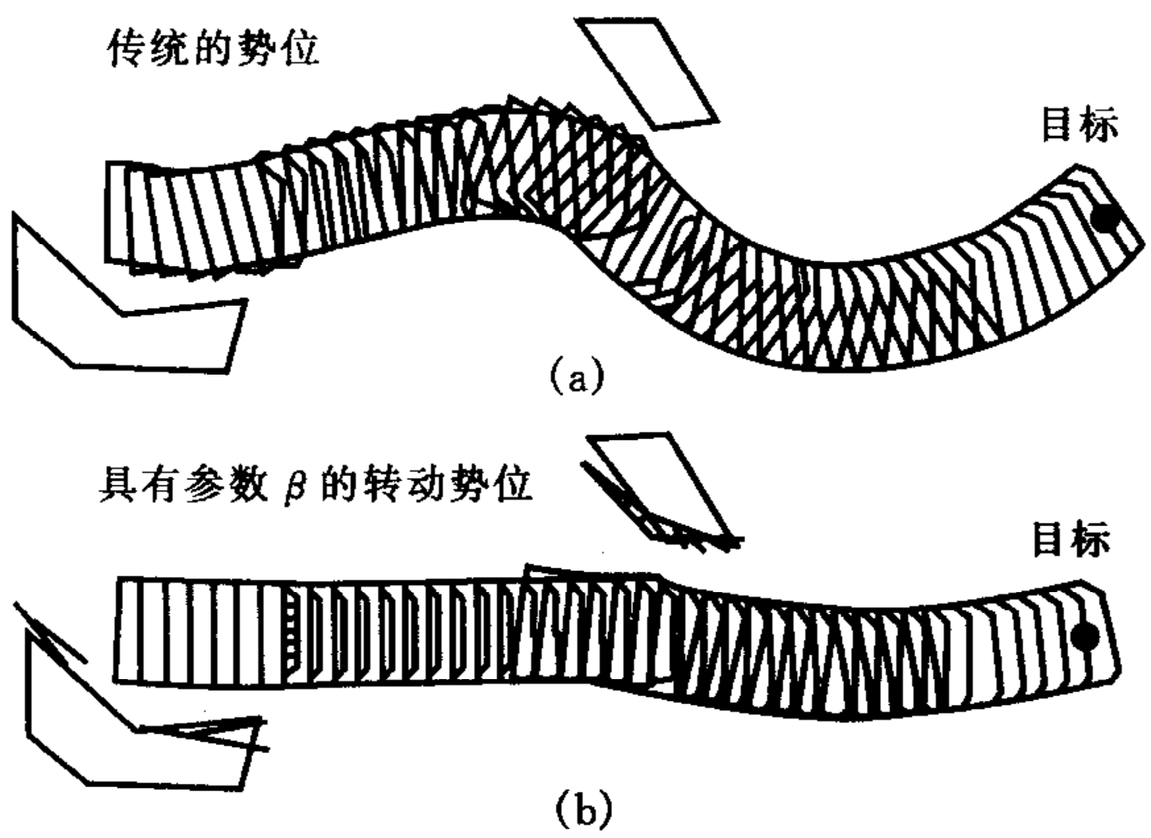


图 6.6 经典势场法与扩展势场法之间的比较。图片由 Raja Chatila[84] 提供

人们已提出了势场法的许许多多的变更和改进，并由移动机器人专家予以实现[67,111]。在大多数情况下，这些变更旨在改善机器人在局部极小中的行为。同时，当机器人必须通过狭窄空间，如门口时，降低震荡和不稳定性的可能性。

势场法非常容易实施，很像在 6.2.1.2 节中描述的野火算法。因此，尽管它存在理论上的局限性，但它在移动机器人应用中已经成为一种普通的工具。

至此,我们完成了移动机器人学中最普通的路径规划技术的简短概述。当然,随着机器人复杂性的增高(例如大自由度的非完全性),而且,特别当环境动力学变得更重要时,上述的路径规划技术就不再适合于解决全方位问题。但对于在大的平坦地形中移动的机器人,机器人学家所用的机动性决策技术常常属于上述范畴之一。

但是,路径规划器可以只考虑环境中机器人已**提前**知道的障碍物。在路径执行期间,由于地图不精确或动态环境,机器人的实际传感器值可能与期望的值不符。所以,根据实际的传感器值,机器人实时地修改它的路径是非常关键的。这就是我们下面将要讨论的机器人**避障**能力。

6.2.2 避障

局部的避障着重于在机器人移动期间,当接到传感器通知时改变机器人的轨迹。最终的机器人运动既是机器人当前的或最近传感器读数的函数,又是机器人目标位置及距目标的相对位置的函数。下面提出的避障算法,不同程度上依赖于全局地图的存在性和相对于地图机器人定位的精确知识。尽管这些算法各不相同,由于机器人局部传感器读数在机器人未来的轨迹中起着重要的作用,下面的所有算法都可以称为避障算法。我们首先提出成功地用于移动机器人学的最简单的避障系统。Bug 算法代表了一种只用最近机器人传感器数值的技术,除了当前传感器数值之外,机器人只需要关于目标方向的近似信息。此后,我们提出更复杂的算法,把当前传感器的历史数据、机器人的运动学和动力学都考虑在内。

6.2.2.1 Bug 算法

Bug 算法[101,102]大概是人们能想象到的最简单的避障算法。其基本思想是在机器人路途中,跟踪各障碍物的轮廓,从而绕开它。

用 Bug1 算法时,机器人首先完全地围绕物体,然后从距目标最短距离的点离开(图 6.7)。当然这种方法效率很低,但可保证机器

人会到达任何可达的目标。

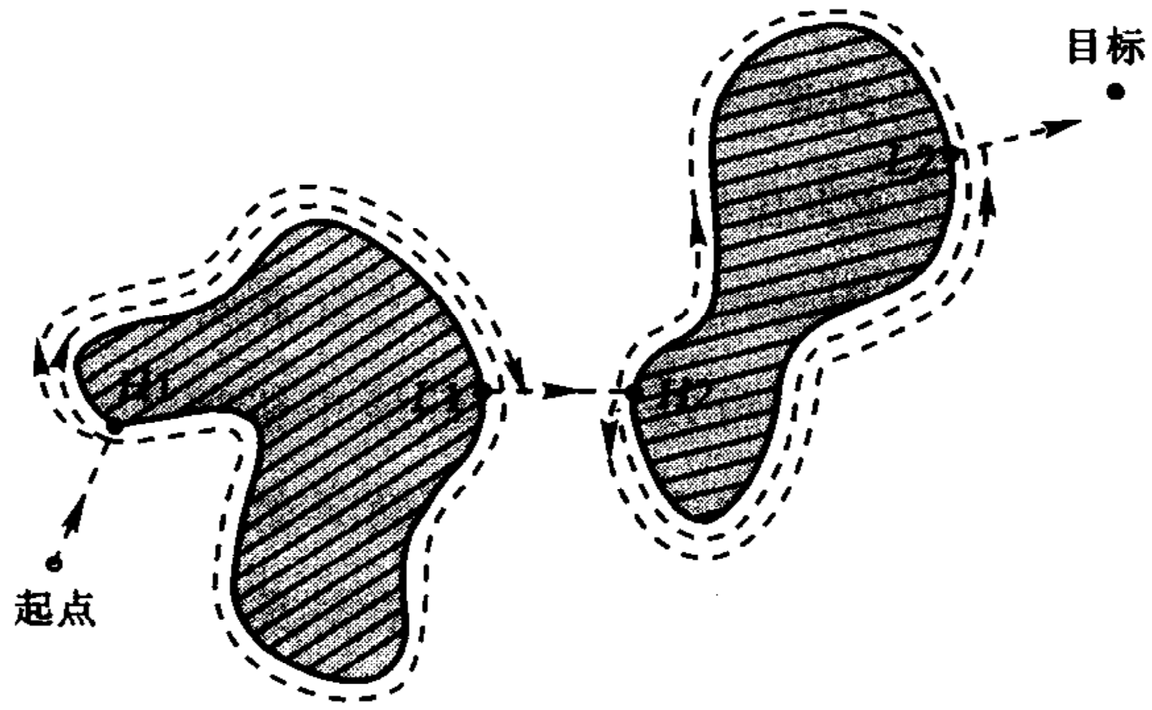


图 6.7 Bug1 算法，H1、H2 为到达点，L1、L2 为分离点[102]

用 Bug2 算法时，机器人开始跟踪物体的轮廓，但当它能直接移动至目标时，就立即分离。一般而言，这个改进的 Bug 算法，如图 6.8所示，具有非常短的机器人行走总路径。但人们仍可构造出使 Bug2 算法无效的情况(即非最优)。

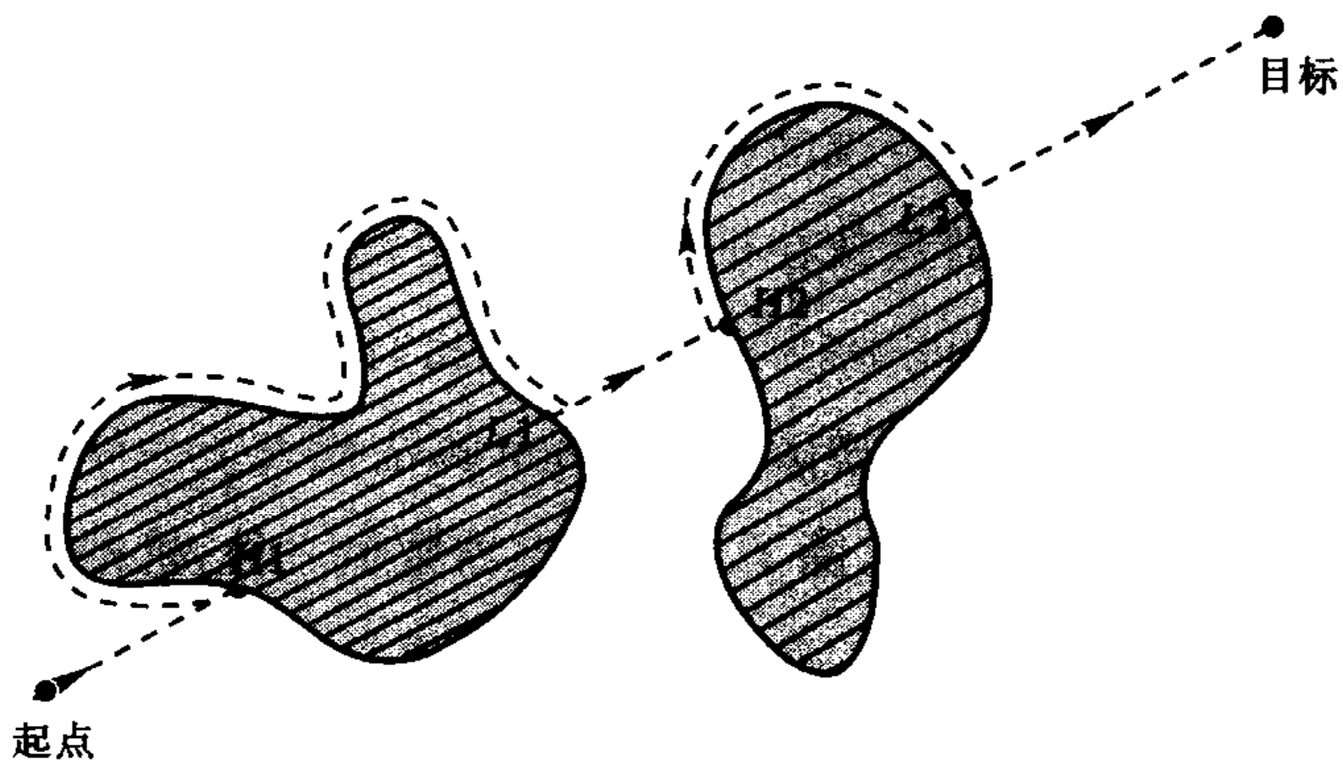


图 6.8 Bug2 算法，H1、H2 为到达点，L1、L2 为分离点[102]

Bug 算法存在许多变种和扩展。我们再提出一个正切 Bug 算法[82]，该方法增加了距离感知和称作局部正切图(local tangent graph，LTG)的局部环境表示。使用 LTG，机器人不仅能更有效地移向目标点，而且当顺着障碍轮廓并切换回到以前目标搜索时，机器

人也可以沿捷径行走。在很多简单的环境中,正切 Bug 算法可得到全局最优的路径。

实际应用:Bug2 算法举例 由于 Bug2 算法的普遍性和简单性,我们利用该技术的变种,给出避障的一个特殊例子。考虑图 6.8 中机器人所采取的路径,我们可以根据两个状态来表征机器人的运动:一个涉及朝向目标点的移动;另一个是围绕障碍物轮廓的运动。我们称前一种状态为 GOALSEEK(目标搜索),后一种状态为 WALLFOLLOW(沿墙壁)。对这两种状态的各个状态,如果我们可以把机器人的运动描述为是传感器值和目标相对方向的函数,并且我们能够描述机器人何时应该在两个状态之间进行切换,那么我们就有 Bug2 算法实用的实现方法。下面的伪代码提供了这种分解的高级控制程序:

```
public void bug2(position goalPos){
  boolean atGoal = false;

  while( ! atGoal){
    position robotPos = robot.GetPos(&sonars);
    distance goalDist = getDistance(robotPos,goalPos);
    angle goalAngle = Math.atan2(goalPos,robotPos) - robot.
GetAngle();
    velocity forwardVel,rotationVel;

    if(goalDist < atGoalThreshold){
      System.out.println("At Goal");
      forwardVel = 0;
      rotationVel = 0;
      robot.SetState(DONE);
      atGoal = true;
    }
```

```
        else {
          forwardVel = ComputeTranslation(&sonars);
          if(robot.GetState() == GOALSEEK){
            rotationVel = ComputeGoalSeekRot(goalAngle);
            if(ObstaclesInWay(goalAngle,&sonars))
              robot.SetState(WALLFOLLOW);
          }
          if(robot.aetState() == WALLFOLLOW)){
            rotationVel = ComputeRWFRot(&sonars);
            if( ! ObstaclesInWay(goalAngle,&sonars))
              robot.SetState(GOALSEEK);
          }
        }
        robot.SetVelocity(forwardVel,rotationVel);
      }
}
```

在理想情况下，当遇到一个障碍物时，根据哪个方向更有希望成功就会在左壁跟踪和右壁跟踪之间作选择。在这个简单的例子中，我们只有右壁跟踪，这是为教学起见的简化，它在实际的移动机器人程序中应该找不到到路径。

现在我们详细考虑各个余下来的功能。对实际情况而言，让我们考虑一个绕机器人径向装有一个声纳环的机器人。这个想象的机器人将是差动驱动的，所以声纳环有一个清晰的“前端”(与机器人前进方向一致)。而且，机器人接受前面所示形式的运动命令，带有转动速度参数和平移速度参数。对两个差动驱动底盘的各个驱动轮，将这两个参数映射成各单独轮子的速度是一件简单的事情。

根据机器人声纳的读数，有一个条件我们必须定义，ObstaclesInWay()。不论何时，当任何声纳传感器的距离数据在目标方向

(目标方向的 45°范围内)是短时,我们定义该函数为真:

```
private boolean ObstaclesInWay(angle goalAngle, sensorvals
sonars){
    int minSonarValue;
    minSonarValue = MinRange(sonars, goalAngle-(pi/4),
                                    goalAngle+(pi/4));
    return (minSonarValue < 200);
} // end ObstaclesInWay() //
```

注意,不论机器人是跟踪墙还是朝向目标,函数 ComputeTranslation()计算平移的速度。在这个简化的例子中我们定义在机器人近似向前方向中平移速度正比于最大的距离读数。

```
private int ComputeTranslation(sensorvals sonars){
    int minSonarFront;
    minSonarFront = MinRange(sonars, -pi/4.0, pi/4.0);
    if (minSonarFront < 200) return 0;
    else return (Math.min(500, minSonarFront - 200));
} // end ComputeTranslation() //
```

这种方法与在 6.2.1.3 节中描述的势场法具有明显的相似性。的确,通过把机器人当前的距离值处理成力向量,一些移动机器人实现了避障,简单地完成向量加法,确定了行走的方向和速度。另一种方法是,许多人把短距离读数考虑为斥力,再参与向量加法,确定机器人总的运动命令。

当面对距离声纳数据时,确定转向和速度的常用方法就是简单地将机器人的左边和右边的距离读数相减。差值越大,则机器人向较长距离读数的转动速度越快。下面两个旋转函数可用于 Bug2 算法的实现:

```
private int ComputeGoalSeekRot(angle goalAngle){
    if (Math.abs(goalAngle) < pi/10) return 0;
```

```
        else return (goalAngle * 100);
    } // end ComputeGoalSeekRot() //
    private int ComputeRWFRot(sensorvals sonars){
        int minLeft,minRight,desiredTurn;
        minRight = MinRange(sonars, -pi/2, 0);
        minLeft = MinRange(sonars, 0, pi/2);
        if (Math.max(minRight,minLeft) < 200) return (400);
            // hard left turn
        else {
            desiredTurn = (400 - minRight) * 2;
            desiredTurn = Math.inttorange(-400, desiredTurn,
            400);
            return desiredTurn;
        } // end else
    } // end ComputeRWFRot() //
```

注意，对右壁跟踪的情况，当右侧存在开阔的空间时，旋转函数将一般的避障与右转偏移量(bias)结合起来，使之保持在与障碍物轮廓接近的地方。这个解决方案，对实现 Bug2 算法肯定不是最好的方案。例如，在右壁跟踪行动期间，通过局部地绘制障碍物轮廓图并使用 PID 控制回路，墙壁跟踪器可以完成更佳的作业。

尽管这些避障算法经常用在简单的移动机器人中，但它们仍有许多缺点。例如，Bug2 算法方法中没有考虑机器人的动力学，而这一点对于非完全机器人来说特别重要。而且，因为只使用最近的传感器数据，传感器噪声对实际环境的性能指标有严重的影响。下面我们设计其他的避障技术，以克服这些限制中的一个或多个。

6.2.2.2 向量场直方图

Borenstein 和 Koren 一起创造了向量场直方图(vector field histogram，VFH)[43]。他们的早期工作主要集中于势场[92]，由于方

法不稳定且没有能力通过狭窄的通道,所以放弃了该方法。后来,Borenstein 和 Ulrich 一起,拓展了 VFH 算法,得到了VFH+算法[150]和VFH*算法[149]。

对 Bug 类型算法的主要批评之一是:机器人在每一时刻的行为一般只是它最近的传感器读数的函数。当机器人的瞬时传感器读数对机器人避障没有提供足够信息的情况下,它可以导致不希望但不可避免的问题。VFH 技术通过创建围绕机器人环境的局部地图,克服了这些限制。这个局部地图是一个小的占有栅格,如 5.7 节所述,只被较新近的传感器距离数据所占用。为了避障,VFH 产生一个极坐标直方图,如图 6.9 所示。x 轴表示发现障碍物角度 α,y 轴表示根据占有栅格的单元值,在那个方向确实存在障碍物的概率 ρ。

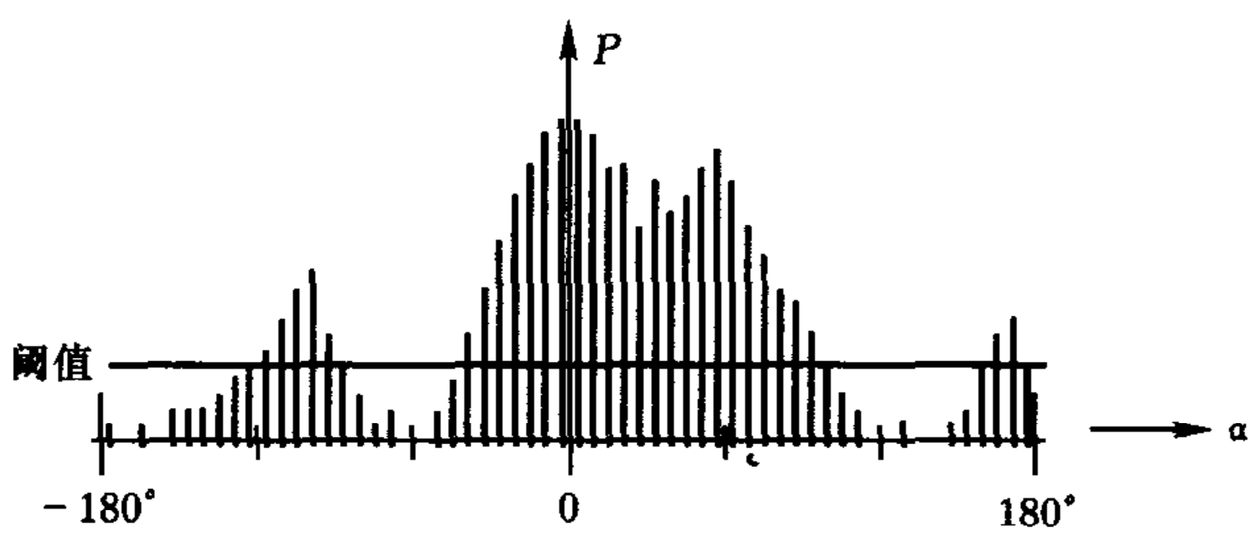


图 6.9 极坐标直方图[93]

由这个直方图,我们可以计算操纵方向。首先确定使车辆通过的足够大的所有开放通路,然后对每一个这样的候选通路计算其费用函数,选择具有最低费用的通路。费用函数 G 有三项:

$$G=a\cdot\text{目标方向}+b\cdot\text{轮子方向}+c\cdot\text{以前方向} \quad (6.11)$$

目标方向=与目标一致的机器人路径;

轮子方向=新方向和当前轮子方向之差;

以前方向=以前所选方向和新方向之差。

计算这些项使得离目标方向大的偏离,导致在"目标方向"项产生大的花费。费用函数 G 中的参数 a、b、c 可调整机器人的行为。

例如，大的目标偏移由大的 a 值来表达。费用函数的完整的定义可参阅参考文献[92]。

在VFH+改进中，基于机器人运动学的限制（例如，调整Ackerman车辆的半径），简化进程之一就是考虑了移动机器人可能轨迹的简化模型。将机器人建模成在圆弧或直线中运动。因此，障碍物阻挡了穿过该障碍物的所有可允许的机器人轨迹（图6.10(a)）。于是形成了被屏蔽的极坐标直方图，在该图中，障碍物被放大，使得所有运动学上被阻塞的轨迹（图6.10(c)）都被适当地予以考虑。

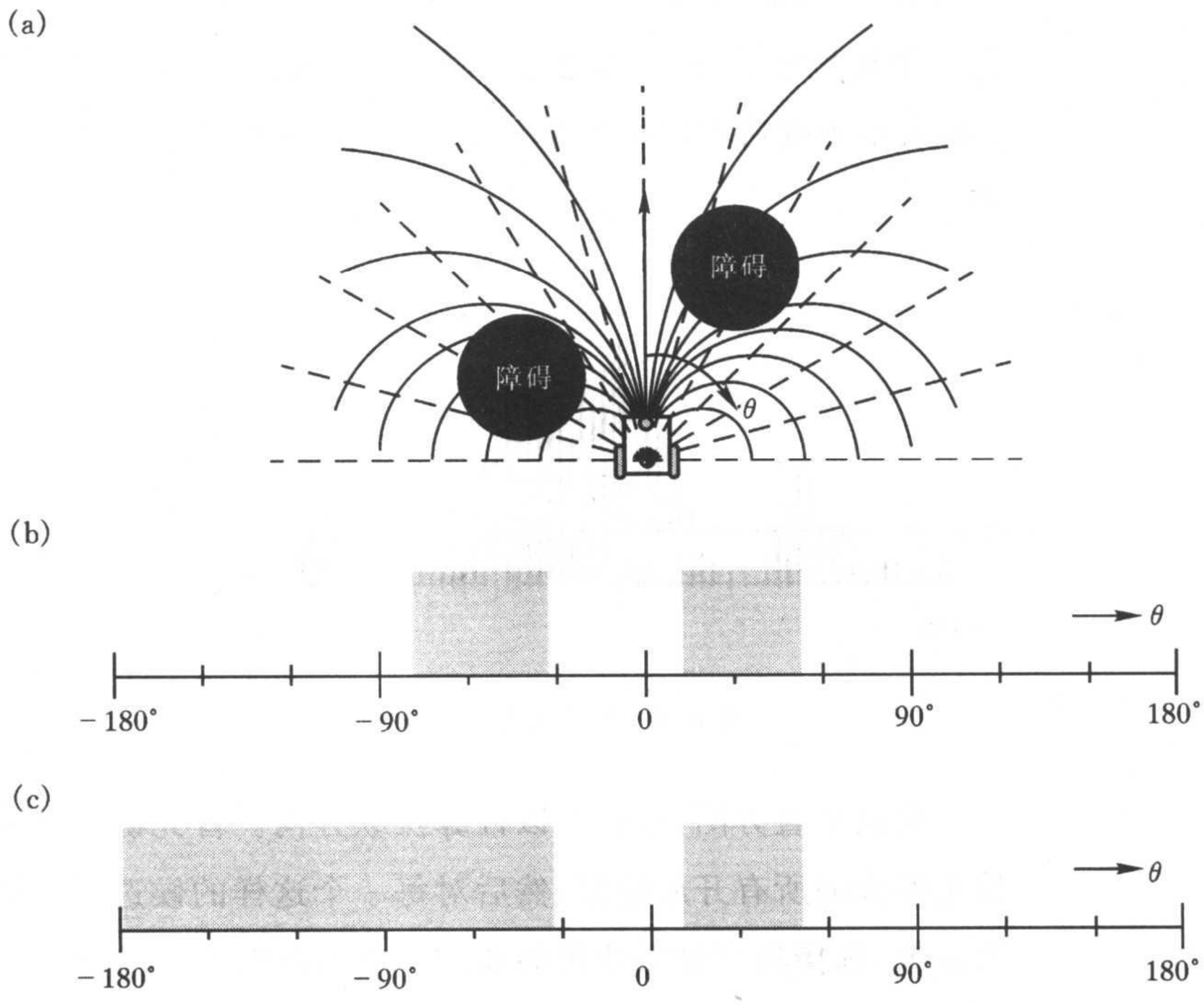


图6.10 被阻塞的方向和最终得到的极坐标直方图举例[54]：(a)机器人与障碍物；(b)极坐标直方图；(c)被屏蔽的极坐标直方图

6.2.2.3 气泡带技术

这种思想是对Khatib和Quinlan提出的弹性带概念[86]中非完全车辆的扩展。最初的弹性带概念只用于完全车辆，所以我们把重

点放在由 Khatib、Jaouni、Chatila 和 Laumod 提出的气泡带扩展技术上[85]。

气泡被定义为围绕给定的机器人配置,其自由空间的最大局部子集,即机械器人可在该局部空间任意方向行走而无碰撞。利用机器人的简化模型连同机器人地图中可用的信息可以产生气泡。当计算气泡大小时,即使用简化的机器人几何特性模型,也有可能考虑到机器人的实际形状(图 6.11)。给定这样的气泡,可以用气泡带或一

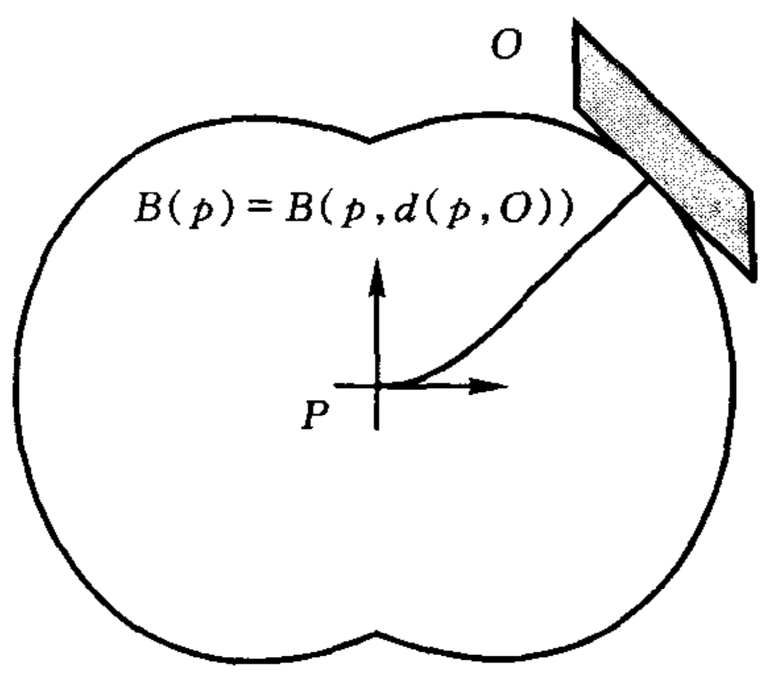


图 6.11 围绕车辆的气泡形状(图片由 Raja Chatila[85]提供)

条沿着从机器人的初始位置到目标位置的轨迹,展示机器人贯穿其路径的期望自由空间(图 6.12)。

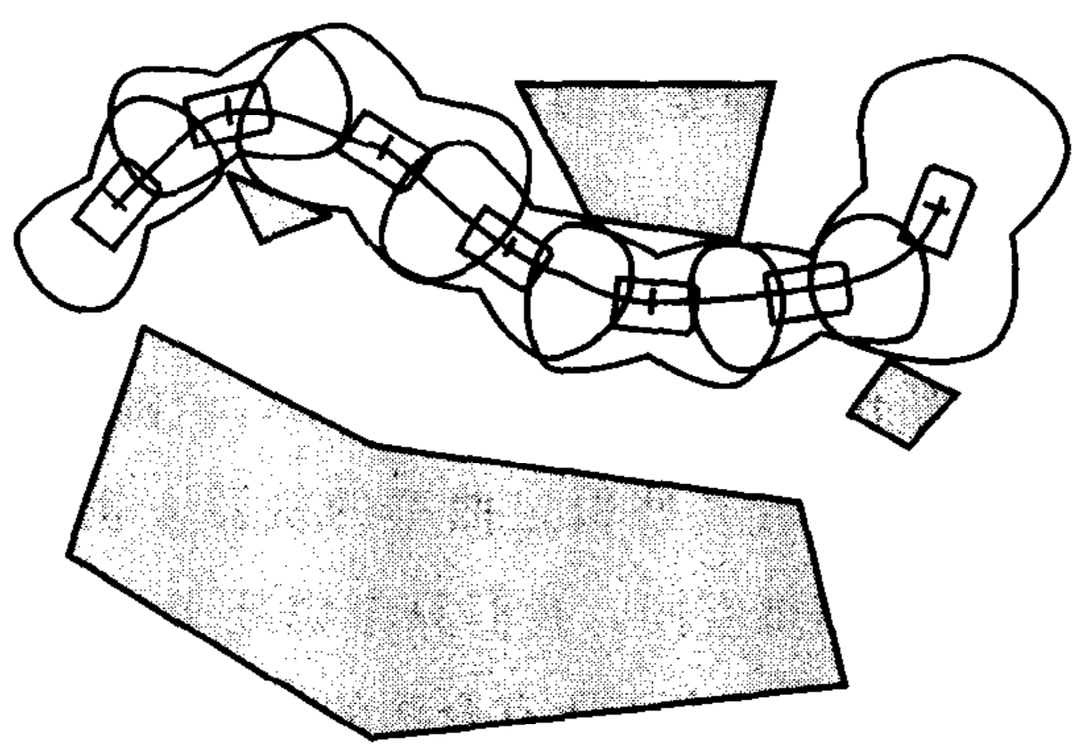


图 6.12 典型的气泡带(图片由 Raja Chatila[85]提供)

显然，计算气泡带需要全局地图和全局路径规划器。一旦已经计算了路径规划器的初始轨迹并计算出气泡带，接着就可对已规划的轨迹作修改。气泡带考虑了已建模物体产生的力和内部的力，这些内部力力图使相邻气泡之间的“缓冲”(能量)最小。在路径执行期间，在使机器人自由空间尽可能地平滑变化这个意义上，上述过程加上最后的平滑操作使得轨迹变得平滑。

当然，到目前为止，这更类似于路径优化而不是避障。气泡带策略的避障方面在机器人运动期间起作用。当机器人得知非预见的传感器值时，利用气泡串模型以一种使气泡带张力最小的方法，使机器人偏转原先的意向路径。

气泡带技术的一个优点是，人们可以考虑机器人的实际尺寸。但正如用离线路径规划技术一样，这种方法只当环境配置事先已很好知道的情况下才最适合于应用。

6.2.2.4 曲率速度技术

基本的曲率速度方法 由 Simmons[135] 提出的曲率速度方法(CVM)能在避障期间考虑到机器人的运动学约束，甚至考虑到某些动力学约束，这是胜过基本方法的一个优点。CVM 一开始将机器人和环境中的物理约束加到速度空间。速度空间由转动速度 ω 和平移速度 v 组成。因此假定，机器人只沿着曲率为 $c=\omega/v$ 的圆弧行走。

我们确认两种类型的约束：由机器人速度和加速度限制造成的约束，具体地说有 $-v_{max}<v<v_{max}$，$-\omega_{max}<\omega<\omega_{max}$；其次是障碍物造成的约束。由于障碍物的位置，它们不允许有某些 v 和 ω 的值。障碍物开始作为笛卡尔栅格中的物体，但随后通过计算机器人位置到障碍物(它沿着某些恒定曲率的机器人轨迹)的距离，如图 6.13 所示，变换到速度空间。我们只考虑处于 c_{min} 和 c_{max} 以内的曲率，因为那个曲率空间包含了所有合法的轨迹。

为达到实时控制的特性指标，障碍物可以由圆形物体近似，并将物体的轮廓划分成几个区段。计算区段端点到机器人的距离，且

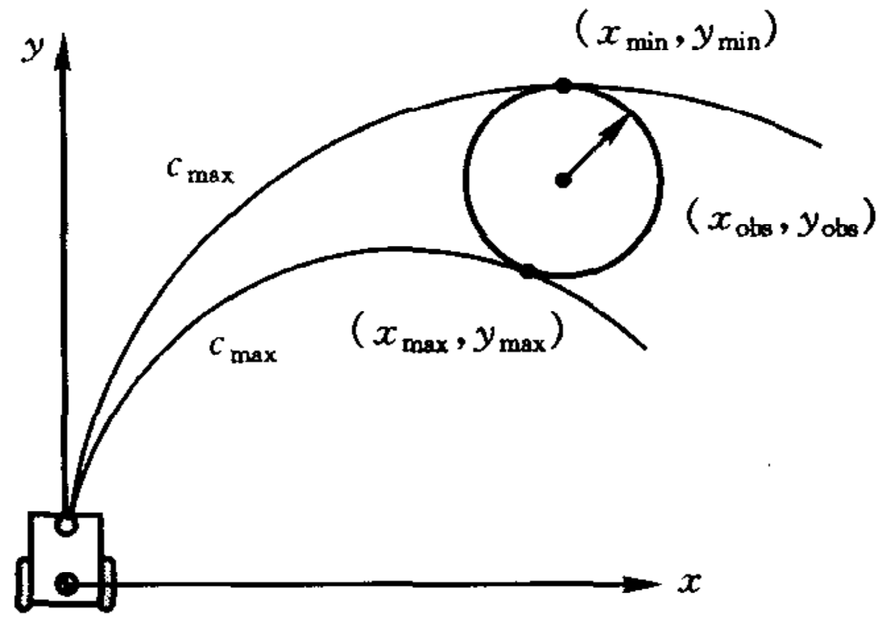


图 6.13 障碍物的切线曲率(引自参考文献[135])

假定端点之间以内的距离函数是恒定的。

目标函数做出最终新速度(v 和 ω)的决策。该目标函数只对履行运动和动力学约束以及障碍约束的速度空间部分进行评估。如果机器人装有多种类型的距离传感器,用笛卡尔栅格作为初始障碍的表示,就能直接进行传感器融合。

CVM 以实用的方式,考虑了车辆的动力学。但这个方法的局限性是障碍物的圆形简化。在某些环境下这是可以接受的。然而,在其他环境中,这种简化可能会导致一系列问题。CVM 还可能产生局部极小,因为系统没有使用先验知识。

道路曲率法 根据 Ko 和 Simmons 对 CVM 缺点的体验,他们提出了 CVM 的改进方法,称之为道路曲率法(lane curvature method,LCM)[87]。CVM 引导机器人通过走廊交叉口有困难。问题来源于机器人只沿着固定的弧线运动的近似性,而在实际中,机器人碰到障碍前可多次改变方向。

LCM 计算一组期望的道路,将到达最近障碍物的道路长度与宽度进行折衷。利用目标函数选择最佳特性的道路。局部的朝向以下述方式进行选择:若机器人还不在最佳道路上,则机器人会转向最佳道路。

实验结果已经展示,它比 CVM 有更好的性能。需要警告的是:

必须仔细选择目标函数的参数以优化系统的行为。

6.2.2.5 动态窗口方法

考虑机器人运动学约束的另一种技术是动态窗口避障法。一个简单但有效的动态模型给这个方法起了名。在文献中描述两个这样的方法，即 Fox，Burgard 与 Thrun 的动态窗口法[69]和 Brock 与 Khatib 的全局动态窗口法[44]。

局部动态窗口法 在局部动态窗口法中，通过搜索一个细心选择的速度空间，考虑了机器人的运动学。速度空间是所有可能组元 (v,ω) 的集合，其中 v 是速度，ω 是角速度。该方法假定，至少在一个时间戳内，机器人只在代表各个组元的圆弧中运动。

给定机器人当前速度，考虑到机器人的加速能力和时间周期，算法首先选择能在下一采样周期内到达的所有组元 (v,ω) 的**动态窗口**。下一步是缩小**动态窗口**，只保留那些保证机器人在碰到障碍之前能够停止的组元。保留的速度被称为容许速度。图 6.14 中描述了一个典型的动态窗口。注意到，动态窗口的形状是矩形的，它遵顺平移和转动的动态能力是独立的这种近似。

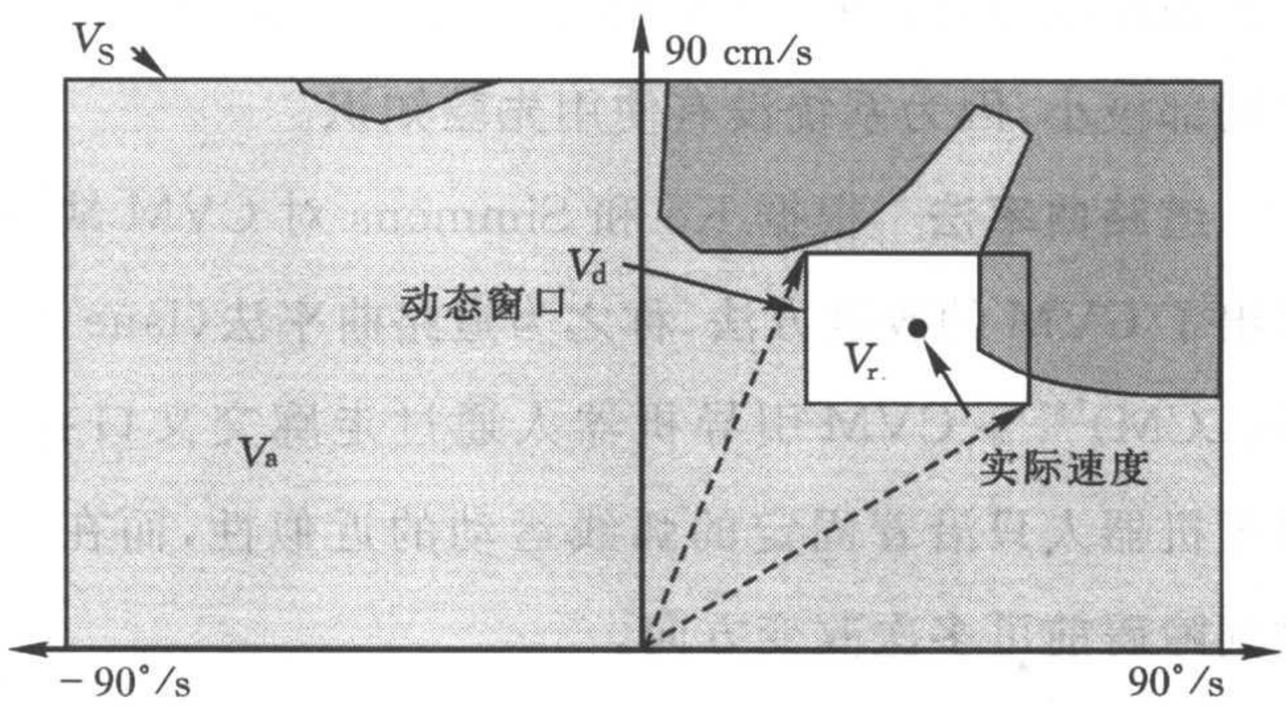


图 6.14 动态窗口方法。矩形窗口表示了可能的速度 (v,ω)，而且与配置空间中的障碍物重叠

将目标函数施加于动态窗口中所有的可容许速度组元，就可选择新的运动方向。目标函数偏爱快速向前运动，与障碍物保持大的

距离,且目标朝向一致。目标函数 O 有以下形式:

$$O=a\cdot \text{heading}(v,\omega)+b\cdot \text{velocity}(v,\omega)+c\cdot \text{dist}(v,\omega) \tag{6.12}$$

heading=向目标位置前进的测量;

velocity=机器人向前的速度→鼓励快速运动;

dist=轨迹中距最近障碍物的距离。

全局动态窗口法 如名称所示,全局动态窗口方法是在上面提出方法上加上了全局思考。这可以由目标函数 O 加上 NF1 算法或野火算法来实现(见 6.2.1.2 节和图 6.15)。回想一下,NF1 方法在占用栅格中,用到目标点的总距离值 L 来标记单元。为了使运算更快,全局动态窗口法只在所选的矩形区进行计算,该区域指引机器人朝向它的目标。如果在这个所选区域的约束内不能达到目标,则扩大区域的宽度并重新计算。

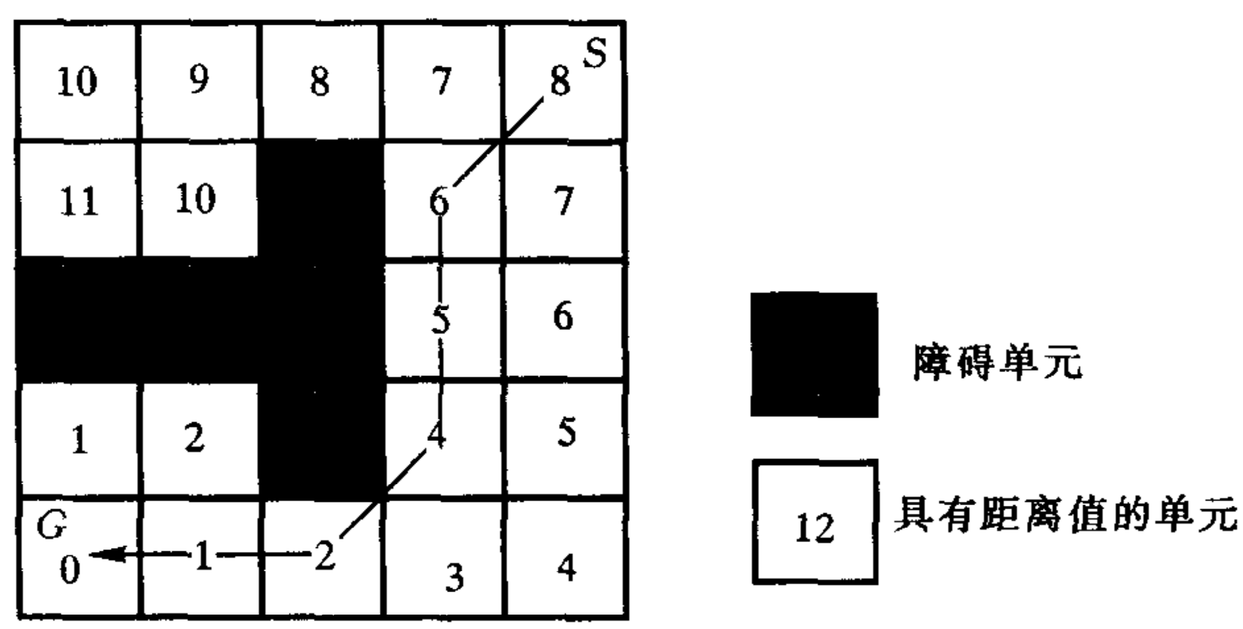


图 6.15 NF1 函数产生的最终路径和距离转换的例子。S 为起点,G 为终点

这就允许全局动态窗口法能获得全局路径规划的优点,而不需完整的先验知识。当机器人在环境中移动时,占用的栅格用距离测量值进行更新,且对每一个更新版本计算 NF1。如果由于机器人被障碍物包围不能计算 NF1,则该方法将退化为动态窗口法。由此保持机器人运动,使之找到一条可能的出路,且可恢复 NF1。

全局动态窗口法保证实时、动态约束、全局思考和高速的最小自由避障。用 450 MHz 单板 PC 的全向机器人已展示了算法的实现。

当占用栅格为 30 m×30 m，具有 5 cm 分辨率时，系统产生大约 15 Hz的周期频率。在测试中，机器人的平均运动速度大于 1 m/s。

6.2.2.6 Schlegel 避障方法

Schlegel[128]提出了一个考虑动力学以及机器人的实际形状的方法。该方法采用笛卡尔栅格表示环境中的障碍物，对原始的激光数据测量和传感器进行融合。利用预先计算的查找表，获得实时性能指标。

如我们前面已描述的许多方法一样，基本假设是：机器人在由圆弧构成的轨迹中移动，圆弧定义为曲率 i_c。给定某曲率 i_c，Schlegel 计算机器人与单个障碍点$[x,y]$之间碰撞的距离 l_i，如图 6.16 所示。因为允许机器人具有任何式样，计算是耗时的，所以结果是预先计算的，并存储在查找表中。

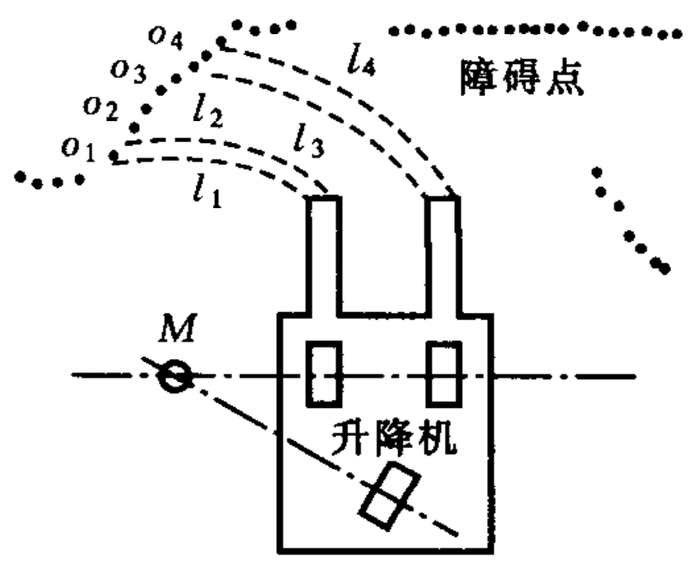


图 6.16 当机器人围绕 M 旋转时，由曲率 i_c 得到的距离值 l_i[128]

例如，对一个差动机器人，搜索空间窗口 V_s 定义为左、右轮所有可能的速度 v_r 和 v_l。给定当前机器人的运动，把 V_s 细化成只在下一时间步内能达到的那些值，就可考虑机器人的动态约束。最后，通过将目标方向、速度、碰撞前的距离进行折衷，目标函数选择最佳的速度和方向。

在测试期间，Schlegel 使用了波前路径规划器。用了两个机器人底盘，一个具有同步驱动运动特性；一个具有三轮运动特性。三轮驱动的机器人令人特别感兴趣，因为它是一个复杂形状的升降机构，

对避障有很大影响。因此,用升降机构的可靠避障示范给人留下了深刻的印象。当然,这个方法的一个缺点是查找表所需的潜在存储器需求。在它们的实验中,作者使用高达 2.5 Mb 的查找表,用 6 m ×6 m笛卡儿栅格,具有 10 cm 分辨率和 323 个不同的曲率。

6.2.2.7 ASL 方法

瑞士联邦理工大学(Swiss Federal Institute of Technology)自主系统实验室(Autonomous System Lab,ASL)为大型展览会[132]开发了一个避障算法[122],要求移动机器人穿过密集人群而移动,还要保证一定的参观者人流。为了得到一个平滑移动而不断地再规划,且在安全情况下能小心地用肘轻轻开拓通路的系统,算法合并了三种方法。这是一个局部的路径规划和避障算法,以从高层取得中途点的形式接受它的输入。图 6.17 给出了该方法的概图。

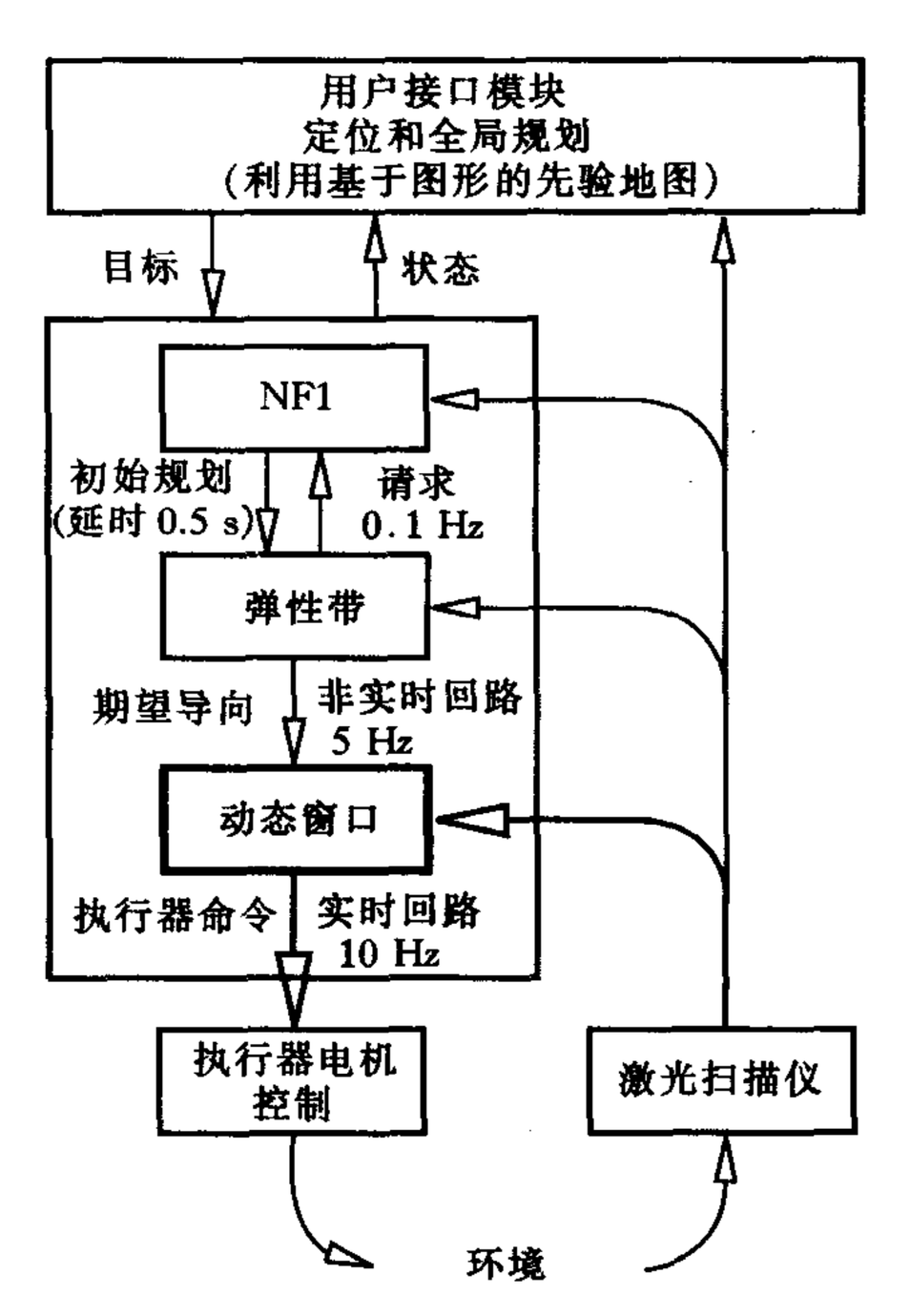


图 6.17 ASL 方法流程图[122]

NF1 执行局部路径规划，利用展览会上所用的八角形机器人在大多时间内能立即旋转的事实，产生的路径被转换成不考虑运动学的弹性带。这样尽可能简单地保持路径的更新。然后，用增强的动态窗口照顾机器人沿着路径移动。

6.2.2.8 接近图

在避障算法中，力图填充模型逼真度的空隙。可以认为接近图(nearness diagram，ND)[107]与 VFH 具有某些相似性，但化解了它的几个缺点，特别是在非常杂乱的空间中。它同样被用在文献[108]中，且考虑了更精密的几何特性、运动学和动力学约束。实现的办法是：把问题分解，产生具有单一约束最有希望的行走方向的一个圆形机器人。然后，使该机器人与它的运动学和动力学约束相适应。如果不是圆形，接着就修正机器人形式(在最初的论文只支持矩形形式)。在文献[110]中，该方法加入了全局推理，称之为全局接近图(global nearness diagram，GND)，这与 DWA 扩展成 GDWA 有些类似，但它基于工作空间的表示，而不是配置空间；且除了障碍物信息外，还更新自由空间。

6.2.2.9 梯度法

认识到当前的计算机技术能对波前传播方法进行快速重复计算，于是梯度法[89]将基于栅格的全局路径规划方程化。它考虑了与障碍物的接近度，并允许在栅格任何给定点上产生梯度方向的连续插补。NF1 方法是所提出算法的一个特殊情况，它在每一时间步计算导航函数，并用得到的梯度信息驱动机器人在平滑路径上走向目标。并且，除非需要，不会擦过障碍物。

6.2.2.10 加上动态约束

在上述讨论的大多数避障方法中，力图强调动态模型的缺乏。在文献[109]中 Minguez、Montano 和 Khatib 提出了一种新的空间表示法。自-动态空间(ego-dynamic space)同样可应用于工作空间和配置空间方法中，它将障碍物变换成距离，而该距离依赖于基本避

障方法的制动约束和采样时间。结合已提出的空间窗口法(PF)来描述加速能力,连同 ND 和 PF 法一起测试了 Minguez 等方法,圆形的完全机器人给出满意的结果,计划把它扩展到非完全、非圆形的体系结构中。

6.2.2.11 其他方法

上面描述的方法是一些最普通的参考避障系统。然而,在移动机器人学范畴,还有众多其他的避障技术。例如,由 Tzafestas 和 Tzafestas[148]对模糊和神经模糊避障的方法提出了综述。受自然界的启发,Chen 和 Quinn[56]提出了一种生物学方法。在该方法中,它们复制了蟑螂的神经网络。然后该网络被用于四轮车辆模型中。

李雅普诺夫函数(Liapunov functions)形成了一个著名的理论,它可以用于证明非线性系统的稳定性。在 Vanualailai,Nakagiri 和 Ha[153]的论文中,李雅普诺夫函数被用于在已知环境中移动的两点质量的控制策略。所有的障碍物均被定义为具有精确位置和圆形的反目标。当建立系统的控制规律时,就使用反目标。然而,就我们所知,这个复杂的数学模型还未在现实环境的机器人上做过测试。

6.2.2.12 综述

表 6.1 给出了避障不同的方法的综述。

表 6.1 最常用的避障算法综述(a)

方法		Bug		
		正切 Bug[82]	Bug2[101,102]	Bug1[101,102]
模型保真度	形式	点	点	点
	运动学			
	动力学			
视图		局部	局部	局部
其他要求	局部地图	局部正切图		
	全局地图			
	路径规划器			

续表 6.1(a)

方法		Bug		
		正切 Bug[82]	Bug2[101,102]	Bug1[101,102]
传感器		距离	触觉	触觉
受试机器人				
特性指标	周期时间			
	体系结构			
评注		很多情况下有效，鲁棒	低效率鲁棒	很低效率鲁棒

表 6.1 最常用的避障算法综述(b)

方法		气泡带		向量场直方图(VFH)		
		气泡带[85]	弹性带[86]	VFH *[149]	VFH+[92,150]	VFH[43]
模型保真度	形式	C 空间	C 空间	圆	圆	过分简单化
	运动学	精确		基本	基本	
	动力学			过分简单化	过分简单化	
视图		局部	全局	基本上局部	局部	局部
其他要求	局部地图			直方图栅格	直方图栅格	直方图栅格
	全局地图	多角形	多角形			
	路径规划器	必需的	必需的			
传感器				声纳	声纳	距离
受试机器人		各种	各种	非完全(GuideCane)	非完全(GuideCane)	同步驱动(六角形)
特性指标	周期时间			6...242 ms	6 ms	27 ms
	体系结构			66 MHz，486PC	66 MHz，486 PC	20 MHz，386AT
评注				较少局部最小	局部最小	局部最小，震荡轨迹

表 6.1 最常用的避障算法综述(c)

方法		动态窗口		曲率速度	
		全局动态窗口[44]	动态窗口法[69]	道路曲率法[87]	曲率速度法[135]
模型保真度	形式	圆	圆	圆	圆
	运动学	(完全)	精确	精确	精确
	动力学	基本	基本	基本	基本
视图		全局	局部	局部	局部
其他要求	局部地图		障碍线场	直方图栅格	直方图栅格
	全局地图	C 空间栅格			
	路径规划器	NF1			
传感器		180°FOV SCK 激光扫描器	24 个声纳环,56 个红外环,立体摄像机	24 个声纳环,30°FOV 激光器	24 个声纳环,30°FOV 激光器
受试机器人		完全(圆形)	同步驱动(圆形)	同步驱动(圆形)	同步驱动(圆形)
特性指标	周期时间	6.7 ms	250 ms	125 ms	125 ms
	体系结构	450 MHz,PC	486PC	200 MHz,奔腾	66 MHz,486PC
局部最小,转入走廊	评注	转入走廊	局部最小		局部最小

表 6.1 最常用的避障算法综述(d)

方法		其他				
		梯度法[89]	全局接近图[110]	接近图[107,108]	ALS 法[122]	Schlegel[128]
模型保真度	形式	圆周	圆周(但通用方程)	圆周(但通用方程)	多边形	多边形
	运动学	精确	(完全)	(完全)	精确	精确
	动力学	基本			基本	基本
视图		全局	全局	局部	局部	局部
其他要求	局部地图		栅格		栅格	
	全局地图	局部感知空间	NF1			栅格
	路径规划器	融合			图形(拓扑的)NF1	前波

续表 6.1(d)

方法		其他				
		梯度法[89]	全局接近图[110]	接近图[107,108]	ALS 法[122]	Schlegel[128]
传感器		180°FOV 离传感器	180°FOV SCK 激光扫描器	180°FOV SCK 激光扫描器	2×180°FOV SCK 激光扫描器	360° FOV 激光扫描器
受试机器人		非完全（近似圆弧）	完全(圆)	完全(圆)	差动驱动（八角形，矩形）	同步驱动（圆），三轮（升降叉车）
特性指标	周期时间	100 ms(核心算法：10 ms)			100 ms(核心算法：22 ms)	
	体系结构	266 MHz,奔腾			380 MHz,G3	
	评注			局部最小	转入走廊	允许改变形式

6.3 导航的体系结构

给出了路径规划、避障、定位和感知解释技术，我们如何将所有这些组合成能为现实环境应用的完整的机器人系统呢？一种着手进行的方法是为用户定制和设计一个特殊应用的完整的软件系统，它实现专用的每件事情。在几乎没有什么特征，甚至更少按计划示范的平凡移动机器人的应用情况下，这可能是有效的。但是对于复杂的和长期的移动机器人，必须按有原则的方式，强调移动性的结构问题。**导航体系结构**的研究，就是针对构成移动机器人导航系统的软件模块进行原理性设计的研究。使用设计良好的导航结构有很多具体的优点：

6.3.1 代码重用与共享的模块性

基本的软件工程原理包含软件的模块性，而且相同的通用动机

因素也同样用于移动机器人。但模块性在移动机器人学中更加重要，因为在单个项目的进程中，移动机器人的硬件或其他的物理环境特征可以剧烈地变化，这是大多数传统计算机所没有碰到的难题。例如，原先只使用超声波测距仪的机器人，有人会引入一个 Sick 激光测距仪；或者在一个新的环境里测试现有导航机器人，而该环境中存在机器人传感器探测不到的障碍物，因此需要一个新的路径规划表示方法。

我们希望改变机器人的部分能力，而不引起一连串副作用，这迫使我们再回顾机器人的其他机能。例如，我们愿意保持避障模块完整，即使在特殊的距离传感器套件发生改变时也是如此。在更极端的例子中，甚至当机器人的动力学结构从三轮底盘变为差动驱动底盘时，如果非完全避障模块能保持不变，这将是理想的。

6.3.2 控制定位

在移动机器人的导航中，机器人控制自定位是一个更为关键的问题。基本原因是机器人的体系结构包括多种类型的控制机能，例如避障、路径规划和路径的执行等。在体系结构中，把各机能定位到特定单元，我们就能够使个别的测试和复合控制的原理性策略成为可能。例如，让我们考虑避免碰撞。为了面对机器人软件变化的稳定性以及为了着重确认避障系统已被正确地实现，将所有有关机器人避障过程的软件进行定位是有价值的。在另一极端，机器人要求高级的规划和基于任务的决策在它们的环境中起有用的作用。因此，将这些高级决策软件定位也同样有价值，它使软件在仿真中无遗漏地被测试，因而，即使与物理机器人无直接连接也可被验证。定位的最后一个优点与学习相关。控制定位可以使一种特定的学习算法只被应用于移动机器人整个控制系统中的一个方面。这种有目标的学习可能是学习和传统移动机器人学产生圆满集成的首要策略。

定位与模块化的优点，为使用原理性导航系统体系结构提供了

令人信服的实例。

表征特定体系结构的一种途径是使用机器人软件的分解性。现在有许多令人喜爱的机器人体系结构，特别是当人们考虑人工智能级的决策和低级机器人控制之间的关系时。对这种高级的体系结构的描述，可参阅文献[2,26]。这里，我们集中于机器人的导航能力。为此，有两种分解方法特别有关：时间分解和控制分解。在 6.3.3 节，我们定义了这两种分解方法，然后对**行为**作了介绍，这是实行控制分解的一般工具。然后，在 6.3.4 节中，我们提出了三种型式的导航体系结构，针对各种体系结构描述了一个已实现的移动机器人实例。

6.3.3 分解技术

分解就是识别坐标轴，沿着坐标轴我们可以判断机器人软件划分成多个独立模块是否正确。分解也用作一种方法，将各种不同的移动机器人分类成更定量的类别。**时间分解**对移动机器人操作区分实时与非实时的要求。**控制分解**是在移动机器人体系结构内，辨识不同的控制输出，联合起来获得移动机器人物理动作的方法。下面我们来详细描述这两种类型的分解。

6.3.3.1 时间分解

机器人软件的时间分解，区分有可变的实时要求和无实时要求的过程。图 6.18 描述了导航的一般时间分解。图中，最实时的过程表示在**堆栈**的底部，无实时要求的过程占据最高层。

在这个例子中，最底层必须用有保证的快周期时间进行，如 40 Hz带宽。相反，准实时层可捕获要求响应时间为 0.1 秒的过程，具有大的、可容许的最坏情况的单个周期时间。战术层可以表示决策，它影响机器人的即时动作。因此，它承受着某些时间约束。而策略层或离线层表示影响机器人长期行为的决定，对模块的响应时间很少有时间限制。

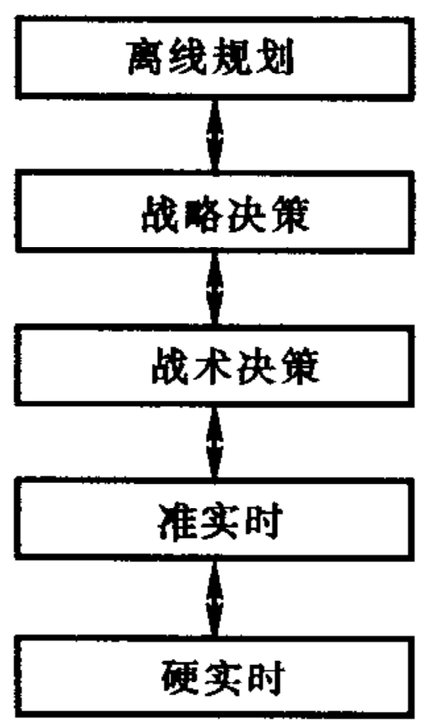


图 6.18 导航体系结构中一般的时间分解

四个重要的、互相关联的趋势与时间分解相关。这些趋势并不是固定的，也存在例外。但是，这些时间分解一般性质是有启发作用的。

传感器响应时间 一个特殊模块的传感器的响应时间，可以被定义为基于传感器事件的获取和模块输出相应变化之间的时间总量。随着图 6.18 层次向上移动，传感器的响应时间趋向增加。对于最底层的模块，传感器响应时间通常只受到原处理器和传感器速度的限制。而对于最高层的模块，传感器响应时间可能受慢的和谨慎决策过程的限制。

时间深度 时间深度是一个应用于时间窗口的有用概念，它影响时间向前和时间向后的模块输出。**时间范围**描述了在选择输出期间，模块所用的预见(时间)量。**时间记忆**描述了传感器输入的历史时间跨度，它被模块用来确定下一个输出。最底层模块倾向于在两个方向上都有很小的时间深度，而最高层模块的慎思过程使用大的时间记忆，并根据它们长期的后果考虑动作，注重大的时间范围。

空间局域性 与时间跨度密切相关，层次的空间影响随模块从底层向高层移动剧烈增加。实时模块倾向于控制轮子的速度和方位，控制空间上的局部行为。高层的策略决策很少或不干涉局部位置，但告知未来的全局位置。

背景特异性 模块使得决策不仅是它即时输入的函数，而且像其他变量所据有的那样，也是机器人背景的函数，诸如环境的机器人表示。最底层模块倾向于直接产生输出作为即时传感器输入的结果，很少使用背景，因此对背景比较不敏感。最高层模块倾向于展示很高的背景特异性。在策略制定时，尽管给定相同的传感器数值，但根据其他背景参数输出可能是完全不同的。

图 6.19 是展示这些趋向的一个例子，它表示了一个简单的导航

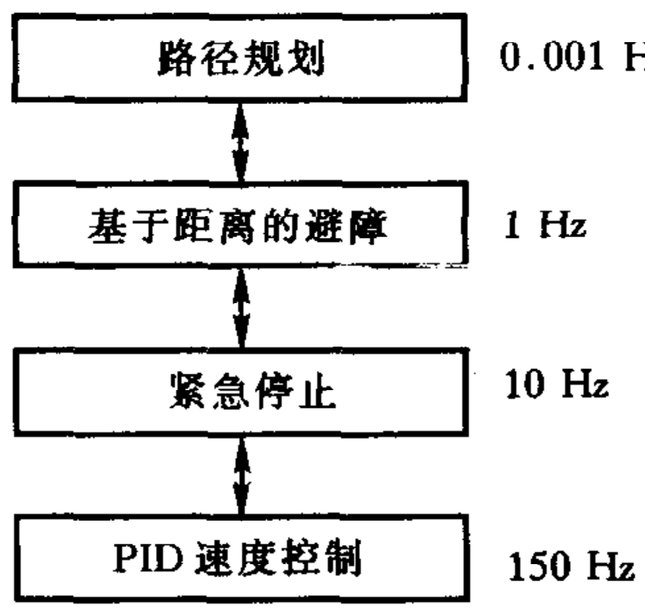


图 6.19 简单导航机器人采样的四级时间分解。右边的列表示出各模块的实际带宽值

体系结构，其时间分解为 4 个模块。在最低层，PID 控制环路提供反馈，控制电机的速度。紧急停止模块使用短距离的光学传感器和缓冲器，当它预测到快要发生碰撞时，切断电机电流。机器人动力学知识意味该模块比 PID 模块有更大的时间范围。接下来的模块使用较长距离的激光测距传感器，辨识机器人正前方的障碍物，并纠正路线偏差。最后，路径规划模块接受机器人的起始位置和目标位置，产生执行的初始轨迹。根据机器人沿着路径所搜集到的障碍物，其轨迹发生变化。

注意到模块的周期或者带宽在相邻模块间按数量级变化。在实际导航体系结构中，这种巨大的差别是很常见的。因此在移动机器人导航体系结构中，时间分解倾向于获取有重要意义的变化轴。

6.3.3.2 控制分解

时间分解是基于软件模块的时间行为进行区分，而控制分解则是识别各模块输出对整体机器人控制输出的贡献。提出控制分解，要求评估者了解离散系统的表示和分析的基本原则。离散系统理论和数学描述的明晰介绍，可参考文献[17,71]。

我们认为机器人算法和物理机器人的实况(即机器人的形状和它的环境)是整个系统组成的成员，我们希望检验系统的连接性。系统 S 由一组模块 M 组成，各个模块 m 经过输入和输出与其他模块相连。如系统是**闭合**的，则意味每一个模块 m 的输入是 M 中一个或多个模块的输出。各模块精确地有一个输出和一个或多个输入，一个输出可以连到任何数目的其他模块的输入。

我们再命名 M 中一个特殊的模块 r，以表示物理机器人及其环境。通常，我们用 r 表示物理对象，机器人算法对此对象产生影响，并由此使机器人算法获得感知输入。模块 r 含有一个输入和一个输出线。r 的输入代表物理机器人完整的动作指标；r 的输出表示输到机器人的完整的感知输出。当然，物理机器人可以有多个自由度，等效地，有多个分立的传感器。但对这个分析，我们只设想完整的输入/输出向量，因此 r 简化为仅一个输入和一个输出。为了简单起见，我们把 r 的输入当做 O，把机器人的传感器数据当做 I。从控制系统的其余部分的观点来看，为了表述我们对 I 和 O 的选择，机器人传感器数据值 I 是输入，机器人的动作 O 是输出。

通过构成机器人算法的系统部分，控制分解区分不同类型的控制通路。在图 6.20 所示的一个极端例子中，我们可以认为它是一个理想的直线或顺序控制通路。

图 6.20 纯串行分解例子

这种串行系统，利用所有相关模块的内部状态以及机器人的感知 I，以顺序方式计算下一个机器人的动作 O。纯串行的结构具有可预测性和可证实性的优点。由于各模块的状态和输出完全依赖于它从上游模块所接收的输入，所以整个系统，包含机器人，是一个单独的、组织良好的回路。因此，整个系统的行为可以用熟知的离散前向仿真方法来评价。

图 6.21 描述了与纯串行控制极其相反的并行控制结构。因为

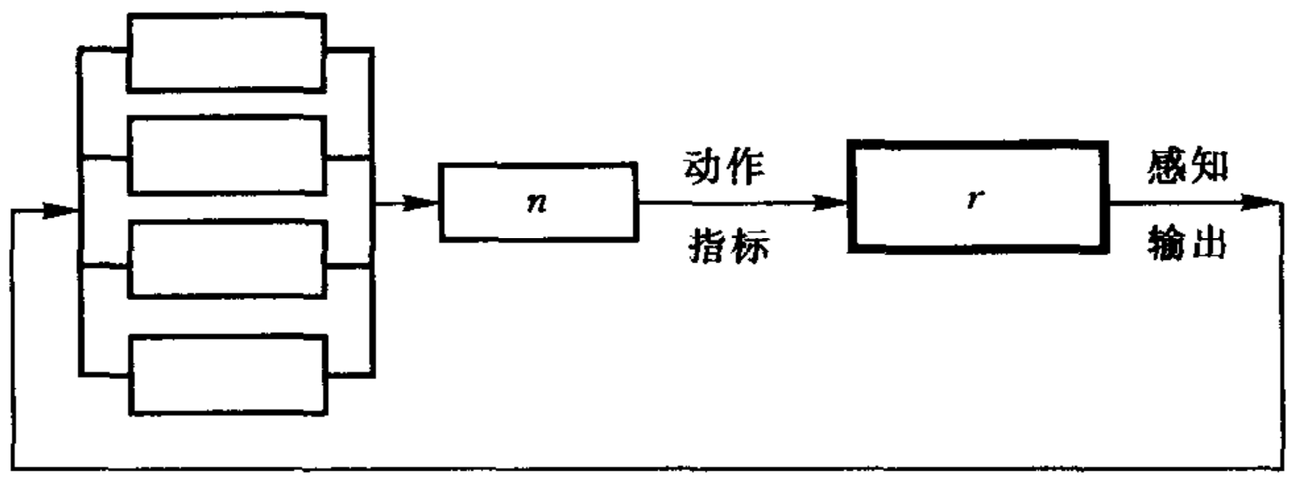


图 6.21 纯并行分解的例子

我们选择将 r 定义为只有一个输入的模块，这个并行系统包括了一个特殊模块 n，它向 r 的消耗提供单一的输出。直观上来说，全并行系统向多个模块同时分发系统控制输出 O 所承担的责任。在纯顺序系统中，控制流是通过一串模块的线性序列。这里，控制流包含**组合**步骤，在这个步骤中，多模块的结果可以以任何方式影响 O。

因此控制的并行化产生了一个重要问题：各组成模块的输出如何通知有关 O 值的总的决策？一个简单的组合技术是时间切换。在这种情况下，它被称作**并行切换**，即系统具有并行分解。但在任何特殊的时刻，输出 O 可以从属于一个特定模块。当然，在各相继的时刻，值 O 依赖于不同的模块，但是 O 的瞬时值总是根据单个模块的功能可以被决定。举例来说，假定机器人有一个避障模块和路径跟踪模块，一个切换控制的实现可能包括：只要机器人距所有可感知障碍的距离超过 50 cm，就执行路径跟踪推荐行为；当任意一个传感器报告距离接近 50 cm 时，就采取避障推荐行为。

如果切换比较少，这种切换控制的优点就特别明显。如果能很好地理解各模块的行为，那就容易表征切换控制机器人的行为：它有时避障，有时路径跟踪。如果已独立地测试了各模块，那么就存在切换控制系统也能很好完成任务的可能性。但必须注意两个重要的缺点。第一，如果切换本身是一个高频事件，那么机器人的整个行为可能变得很差。在这种情况下，机器人可能不稳定，即切换运动的模是如此的迅速，以至剧烈地导致既不是路径跟踪又不是避障的行为。切换控制的另一个缺点是当机器人避障时它没有路径跟踪的偏置（反之亦然）。因此，在控制**应该**由多个模块混合进行推荐的场合，切换控制方法会失败。

相反，更复杂的**混合式并行**模型容许控制在任何给定时刻在多模块之间共享。例如，在所有时刻，相同的机器人可获得避障模块的输出，把它转换成速度向量，并用向量加法将它和路径跟踪模块的输出组合。因此机器人的输出决不会来源于单一模块的输出，而会由两个模块输出的数学组合产生。混合式并行控制比切换控制更普通，因此，它也是一个要使用得当、更具挑战性的技术。尽管使用切换控制，最差的行为却由不合适的切换行为而引起。但在混合控制中，机器人的行为也可以十分差，甚至更容易如此。正像在决定突然旋转以避障时，组合的多个向量可以产生笔直朝前很坏决策一样，在数学上，组合的多个推荐行为不保证有全局优越的结果。因此，在混合并行控制的实施中，必须非常小心地构造混合的方程和各个模块的指标以产生有效的混合效果。

切换和混合式并行结构二者在基于行为的机器人领域中是普通的。Arkin[2]提出了**电机-模式**结构，在这个结构中，**行为**（即上面讨论的模块）把传感器值向量映射成电机值向量。如在混合式并行系统一样，利用单个行为输出的线性组合，产生机器人算法的输出。相反，Maes[103,104]通过建立**行为网络**产生切换的并行结构。在行为网络中，通过比较和更新各行为的激励水平，可分立地选择行为。

Brooks[45]的包含结构是切换并行结构的另一个例子，虽然该主动模型是通过抑制机制，而不是激励水平进行选择。更详细的讨论见参考文献[2]。

并行控制的总缺点是机器人特性指标的验证可能是极其困难的。因为这种系统常常包括真正并行和多线程实现，错综复杂的机器人-环境交互和要求合适地表示所有可能的模块-模块交互的传感器定时，是困难或不可能仿真的。因此，在并行控制领域中，很多测试都是用物理机器人凭经验进行的。

并行控制一个重要优点是它的仿生特征。复杂的有机组织受益于大规模的真正并行化（比如，人的眼睛），并行控制领域的一个目标就是理解生物学上这种一般性策略，并使它能起杠杆作用以有利于机器人学。

6.3.4 实例研究：分层机器人结构

机器人专家常常把多个模块组合在一起组成机器人结构体系，按这个公共主题，我们已经描述了机器人结构的时间和控制分解。把这种理解记在心里，让我们再回到总的移动机器人导航任务上。显然，机器人的行为在机器人实时层次上（比如，在路径跟踪和避障）起重要作用。在较高的时间层，为了沿着意向路径实现机器人运动，更多的战术任务需要调整行为或模块的激励。更高层也是如此，全局规划器产生路径，它向战术任务提供全局性的预见。

在第1章，我们介绍了功能分解，该分解从信息流的观点出发，展示了移动机器人导航器的多个模块。这里，我们将相关的图重新表示于图6.22。

在这种表示中，弧线表示实时和非实时能力诸方面。例如，避障要求定位模块的少量输入，并且在认知层形成快速的决策，随后在运动控制层执行。相反，PID位置反馈回路越过所有的高层处理，把感知的编码器值直接连到运动控制最低层的PID控制回路。在这种

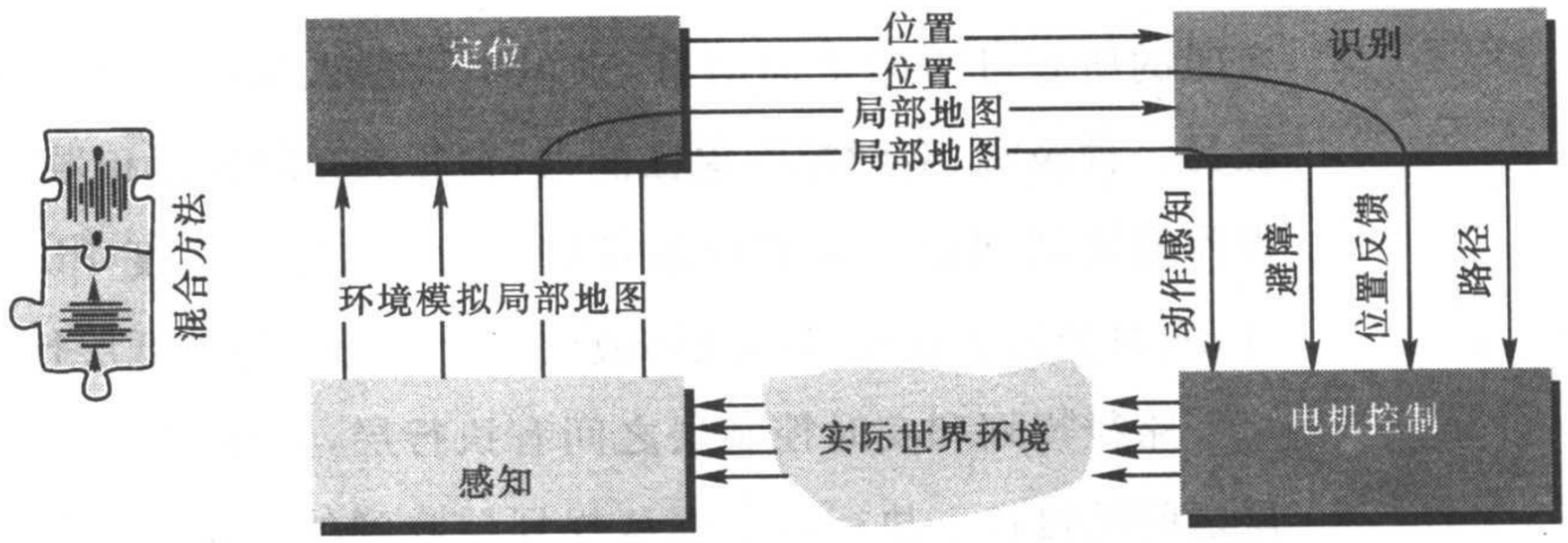


图 6.22 全书应用的基本体系结构举例

表示中，通过四个软件模块的弧线轨迹提供了时间信息。

利用本章的工具，我们现在可从机能的时间分解角度出发提出相同的体系结构。因为我们希望在导航系统中讨论策略、战术和实时处理之间的交互，所以这是特别有用的。

图 6.23 描述了基于 Pell 和其同事所提出的通用分层结构[120]，在设计自主飞行器“**外层空间** I”中采用了该方法。这个图在表示机器人能力的时间分解方面与图 6.19 是相似的。可是，将各模块与相邻模块分离的边界对机器人导航来说是特定的。

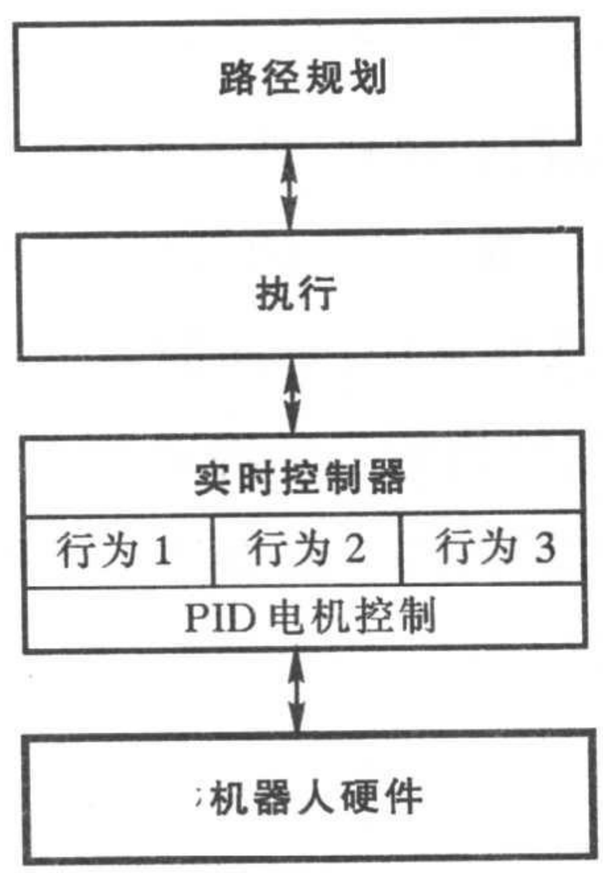


图 6.23 基于时间分解通用的多层移动机器人的导航体系结构

路径规划将移动机器人策略层的决策具体化。路径规划使用非实时的所有可用的全局信息，来识别机器人局部行动的正确顺序。在另一种极端，**实时控制**象征能力，它要求高的带宽和强的传感器—操作器控制回路。在它的最低层，它包括电机速度 PID 回路。此外，实时控制也包括可形成切换或混合并行结构的低层行为。

路径规划和实时控制层之间有**执行**层，负责调节规划层和执行层之间的接口。执行层根据从规划器收到的信息，管理行为的各种激励。执行程序也负责识别错误、保存(将机器人置于稳定状态)、甚至当需要时重启规划器。正是在这个结构中，执行程序包含了所有战术的决策和机器人短期记忆的频繁更新，如同定位和绘图情况一样。

注意到，令人感兴趣的是一般结构和 Shakey 实现的结构有相似性，前者以许多特殊的形式用于当今移动机器人中；后者是最早的移动机器人之一(1969 年[115])。Shakey 具有构成最低结构层的 LLA(低层动作)。各 LLA 的实现，如同今天的机器人一样，它包括了在闭路中使用的传感器值。在它上面，中间结构层包括了 ILA(中间层动作)，它在执行期间，根据感知反馈的要求激励和解除对 LLA 激励。最后，Shakey 的最高层是 STRIPS(斯坦福研究所规划系统)，它提供了全局预测和规划，把一系列任务分发到中间层去执行。

虽然，作为机器人导航的模型，图 6.23 所示的一般结构是很有用的，但在机器人领域中，不同的实现方法可以差别很大。下面，我们给出一般分层结构的三种特别型式，对各种型式至少描述一个现实环境的移动机器人的实施方法。各种机器人结构更广泛的讨论，见参考文献[26]。

6.3.4.1 离线规划

的确，规划和执行的最简单的可能集成就是根本无集成。考虑图 6.24，图中只有两个软件层。在这种导航结构中，执行层没有一个规划器，随它自由处理，但它必须预先包含行走到期望目的地的所

有相关方案。

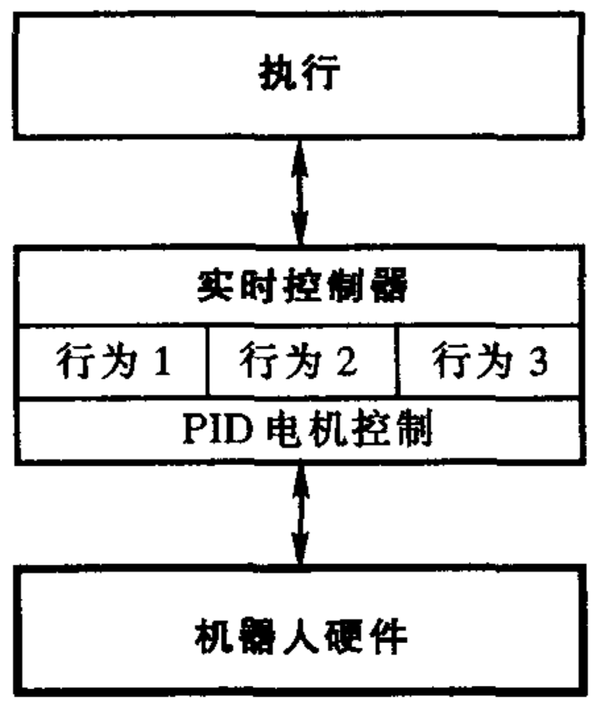


图 6.24 离线规划的两层结构

完全省略规划器的策略当然是极其有限的。把这样的一个机器人移动到一个新的环境中，要求导航系统有一个新的示例。因此这种方法作为导航问题的一般解决方案是无用的。可是，这种机器人系统确实存在，此方法在以下两种情况下可能是有用的：

基于静态路径的应用 在移动机器人的应用中，利用基于路径的导航系统，机器人运行在一个完全静态的环境中。这种情况是可以想象得到的，即分立的目标位置数目是如此之少，以至环境的表示可以直接包含机器人到所有期望目标点的路径。例如，在工厂或者仓库的环境里，机器人跟随埋在地下的导向线可以行走一个单独的环路。在这种工业应用中，当预先编译的一套基于路径的解决方案可以由机器人编程员方便地产生时，路径规划有时是完全不需要的。Chips 移动机器人是博物馆机器人[118]的一个例子，它也使用这种结构。Chips 运行在由有色路标确定的单向环形轨导上。而且，它只有十二个容许停顿的分散位置。由于这种环境模型的简单性，Chips 只含有一个执行层，它直接缓存所要求的路径以便到达各目标位置，不必用一般的地图(用地图路径规划器可以搜索求解路径)。

极端可靠性的要求 不必惊奇，避免在线规划的另一个原因是使系统的可靠性最大。因为规划软件可能是移动机器人软件系统最

复杂部分，又因为至少在理论上规划可能占用与问题复杂性成指数关系的时间，所以在成功的规划上加入硬性的时间约束是困难的，如果不是不可能的话。通过离线计算所有可能的解决方案，工业移动机器人可以用多样性换取有效的定时规划（当然，与此同时牺牲了大量的内存）。在为航天飞行器飞行设计的应急规划中，我们可以看到为此理由而建立离线规划的实际例子。为了不让宇航员在线解决问题，成千上万能想到的问题都在地球上作了假设，并在航天飞行器飞行之前设计了完整的有条件规划。基本目标就是在宇航员开始解决问题之前，提供经历时间总量的绝对上限，以大量的地面时间和文件工作为代价保证实现特性指标。

6.3.4.2 片断规划

离线规划基础信息论的缺点是规划的适应性差。在运行时刻，机器人肯定会碰到提供信息的感知输入，在随后的执行中考虑该附加信息是合理的。在当前移动机器人的导航中，片断规划是最流行的方法。因为它以计算上容易处理的方式解决这个问题。

如图6.25所示，与图6.23的一般结构相仿，结构是三层的。隐

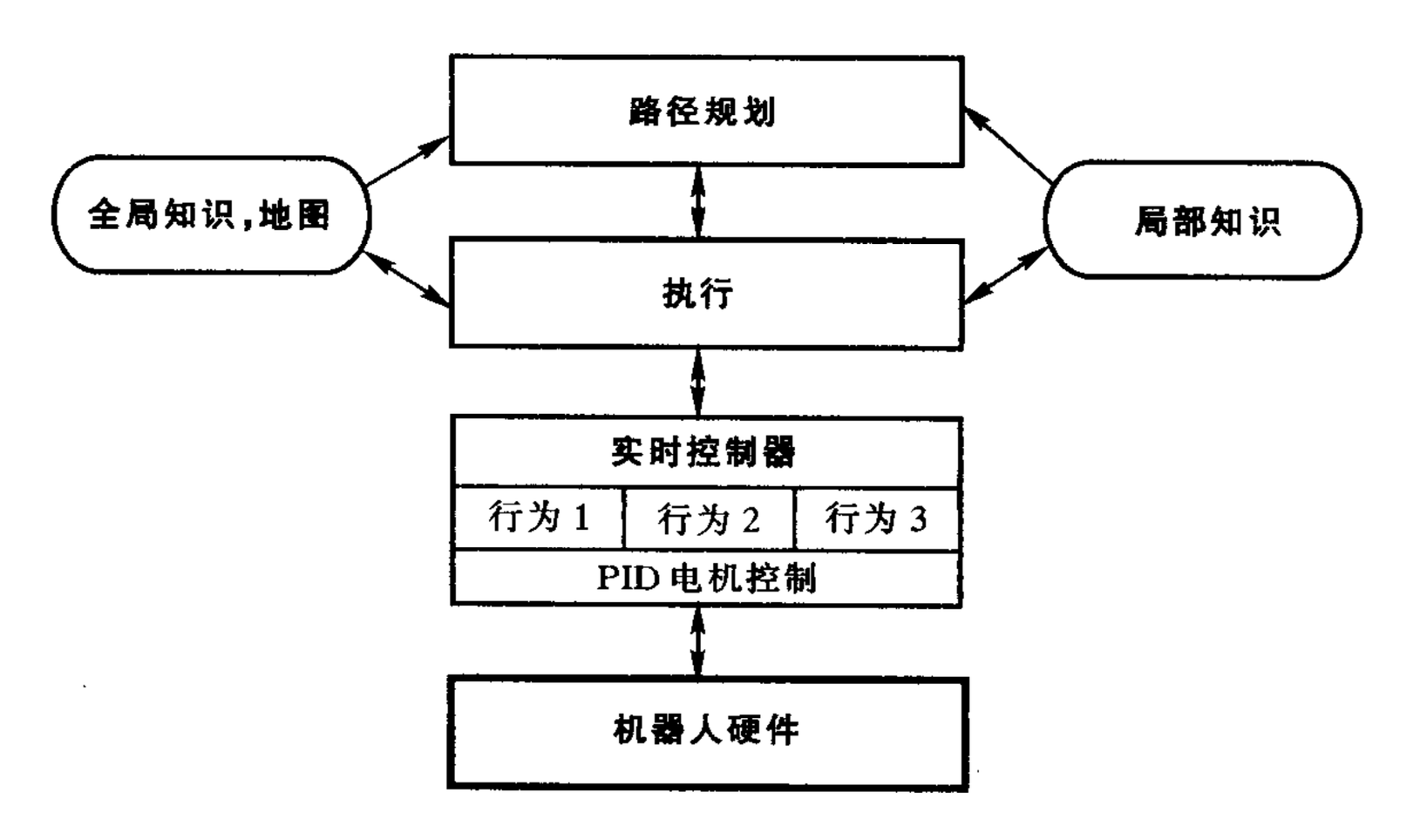


图6.25 三层的片断规划结构

藏在规划器角色后面的直觉如下：规划是计算上密集的，因此规划太

频繁将是严重的缺陷。但执行层处在极好的地位，它辨识何时遇到足够的信息（例如，通过特征提取），可保证策略方向的重大改变。在这样的时候执行层会调用规划器，比如，产生一条新的到目标的路径。

也许触发重新规划最明显的条件是在意向的行走路径上检测到阻塞。例如，在参考文献[129]中，如果几秒钟没前进，则路径跟踪行为返回失败。执行层收到这个失败的通知，修改机器人环境的短期占有栅格的表示，并考虑到局部环境地图的改变，就启动路径规划器。

推迟规划直到获得更多信息，这个普通技术被称为**延期规划**。这个技术在具有动态地图的移动机器人中特别有用，它使机器人移动时更为准确。例如，可以给商用 Cye 机器人一组目标位置。利用它的野火（广度优先）规划算法，该机器人会绘制出仅到最近目标位置的详尽路径，并会执行这个规划。一旦到达这个目标位置，根据运动期间所提取的感知信息，它的地图将会改变。只在这时 Cye 执行层触发路径规划器产生一条从它的新的位置到下一个目标位置的路径。

当在途中遇到未预见的障碍时，伴随更复杂的策略，机器人 Pygmalion 实现了片断规划的结构。当最低层的行为未能做出进展时，执行层通过把机器人旋转 90°再次试探，力图找到一条绕过障碍的路径。这是值得的，因为机器人在运动学上是不对称的，所以将越过一个特殊障碍的路线进行修正，可能在一个方向比另外方向更容易。

为了路径规划，Pygmalion 的环境表示由连续的几何模型和抽象的拓扑网络组成。如果重复地尝试避障清理失败，机器人的执行层将临时切断两个可能阶段之间的拓扑连接，并重新启动规划器，产生一组到目标的新路径点。接下来，利用最近的激光测距仪的数据作为一类局部地图（图 6.25），几何路径规划器将产生一条从当前机器人位置到下一路点的路径。

总之，在移动机器人研究领域中，片断规划结构极为普遍。利用战术执行层快速响应和控制实时机器人运动的行为，它们把响应环境变化的多功能性与新目标结合起来。如图6.25所示，在这种系统中，既有短期的局部地图，又有更为策略的全局地图，这是共性的。在这种双重表示中，执行层的作业部分决定新信息何时集成到局部地图，以及它是否有足够长时间被复制到全局知识库。

6.3.4.3 集成的规划和执行

当然，商业移动机器人的结构必须包含比仅有导航更多的机能。把讨论限制在导航结构的问题上，会导致一开始也许是一个退化解。

图6.26表示的结构看起来与图6.24表示的离线规划结构相

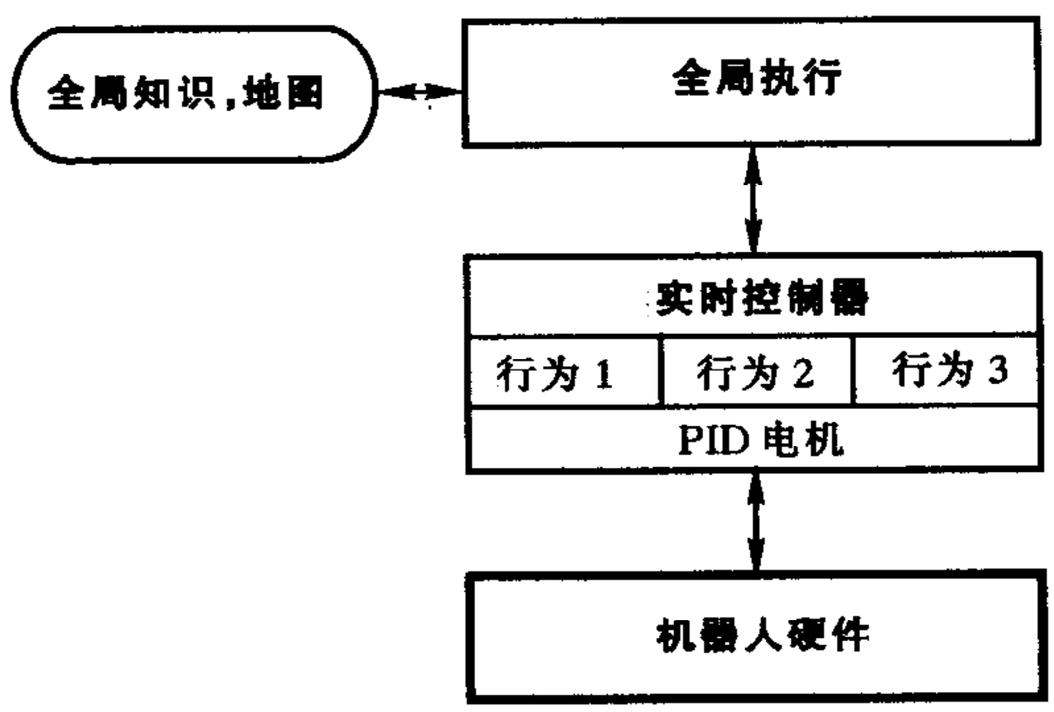


图6.26 一个集成的规划与执行体系结构，在该结构中规划只不过是一个实时的执行步骤(行为)

似，但实际上它远比后者更为先进。在这种情况下，因为在执行和规划器之间不再有时间分解，所以规划层已不存在。规划只是执行层正常活动周期的一小部分。

加速规划使执行时间不再很长的概念似乎是向往的。不过，在特殊环境中使用特殊的算法，已展示了这类规划代价的缩减。让我们考虑Stentz[139]的工作，他为高速行走在部分已知地形的大型越野车辆，设计了一个移动机器人的控制结构。利用计算科学中先进的缓存技术，Stentz优化了野火的路径跟踪算法，称为D*算法，使得

全局路径规划有可能在执行层的基本控制回路内完成。

结果(表示在图 6.26 中)成为局部和全局表示是相同的一种结构。在该结构中,执行层具有内在问题所需的所有的全局规划机能。这个方法的优点是机器人每个循环的行动由全局路径规划器引导。由于考虑了机器人已收集到的所有信息,因而行动是最优的。当然,这个方法在计算上有挑战性,除非处理器的速度更快,不然,在更复杂的环境中将是不切合实际的。随着环境规模的增大,它也存在可应用性的基本限制。但当这种方法用于现实世界场景那样的规模时不会有阻碍。

集成的规划和执行方法 D* 最近的成功强调了这样一个事实:机器人导航结构的设计者不但要考虑到机器人及其环境任务的所有方面,而且还要考虑到处理器和内存技术的发展状态。我们认为移动机器人体系结构的设计确是未来几年革新的活跃领域。从机器人传感器的发明到微处理器速度的提高,所有形式的技术进步就像以前不能想象的策略变成现实一样,都可能催化移动机器人结构新的革命。

参考文献

Books

[1] Adams, M.D., *Sensor Modelling, Design and Data Processing for Autonomous Navigation*. World Scientific Series in Robotics and Intelligent Systems. Singapore, World Scientific Publishing Co. Ltd., 1999.

[2] Arkin, R.C., *Behavior-Based Robotics*. Cambridge MIT Press, MA, 1998.

[3] Bar-Shalom, Y., Li, X.-R., *Estimation and Tracking: Principles, Techniques, and Software*. Norwood, MA, Artech House, 1993.

[4] Borenstein, J., Everet,t H.R., Feng, L., *Navigating Mobile Robots, Systems and Techniques*. Natick, MA, A.K. Peters, Ltd., 1996.

[5] Borenstein, J., Everett, H.R., Feng, L., *Where Am I? Sensors and Methods for Mobile Robot Positioning*. Ann Arbor, University of Michigan. Available at http:// www-personal.engin.umich.edu/~johannb/position.htm.

[6] Breipohl, A.M., *Probabilistic Systems Analysis: An Introduction to Probabilistic Models, Decisions, and Applications of Random Processes*. New York, John Wiley & Sons, 1970.

[7] Bundy, A. (editor), *Artificial Intelligence Techniques, a Comprehensive Catalogue*. New York, Springer-Verlag, 1997.

[8] Canudas de Wit, C., Siciliano, B., and Bastin G. (editors), *Theory of Robot Control*. New York, Spinger-Verlag, 1996.

[9] Carroll, R.J., Ruppert, D., *Transformation and Weighting in Regression*. New York, Chapman and Hall, 1988.

[10] Cox, I.J., Wilfong, G.T. (editors), *Autonomous Robot Vehicles*. New York, Spinger-Verlag, 1990.

[11] Craig, J.J., *Introduction to Robotics: Mechanics and Control*. 2nd edition. Boston, Addison-Wesley, 1989.

[12] de Silva, C.W., *Control Sensors and Actuators*. Upper Saddle River, NJ, Prentice Hall, 1989.

[13] Dietrich, C.F., *Uncertainty, Calibration and Probability*. Bristol, UK, Adam Hilger, 1991.

[14] Draper, N.R., Smith, H., *Applied Regression Analysis*. 3rd edition. New York, John Wiley & Sons, 1988.

[15] Everett, H.R., *Sensors for Mobile Robots, Theory and Applications*. New York, Natick, MA, A.K. Peters, Ltd., 1995.

[16] Faugeras, O., *Three-Dimensional Computer Vision, a Geometric Viewpoint*. Campridge MIT Press, MA, 1993.

[17] Genesereth, M.R., Nilsson, N.J., *Logical Foundations of Artificial Intelligence*. Palo Alto, CA, Morgan Kaufmann, 1987.

[18] Haralick, R.M., Shapiro, L.G., *Computer and Robot Vision, 1+2*. Boston, Addison-Wesley, 1993.

[19] Jones, J., Flynn, A., *Mobile Robots, Inspiration to Implementation*. Natick, MA, A.K. Peters, Ltd., 1993. [also available in German and French].

[20] Kortenkamp, D., Bonasso, R.P., Murphy, R.R. (editors), *Artificial Intelligence and Mobile Robots; Case Studies of Successful Robot Systems*. Cambridge, MA, AAAI Press / MIT Press, 1998.

[21] Latombe, J.-C., *Robot Motion Planning*. Norwood, MA, Kluwer Academic Publishers, 1991.

[22] Lee, D., *The Map-Building and Exploration Strategies of a Simple Sonar-Equipped Mobile Robot*. Cambridge, UK, Cambridge University Press, 1996.

[23] Leonard, J.E., Durrant-Whyte, H.F., *Directed Sonar Sensing for Mobile Robot Navigation*. Norwood, MA, Kluwer Academic Publishers, 1992.

[24] Manyika, J., Durrant-Whyte, H.F., *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*. Ellis Horwood, 1994.

[25] Mason, M., *Mechanics of Robotics Manipulation*. Cambridge, MA, MIT Press, 2001.

[26] Murphy, R.R., *Introduction to AI Robotics*. Cambridge, MA, MIT Press, 2000.

[27] Nourbakhsh, I., *Interleaving Planning and Execution for Autonomous Robots, Norwood, MA*. Kluwer Academic Publishers, 1997.

[28] Raibert, M.H., *Legged Robots That Balance*, Cambridge, MA, MIT Press, 1986.

[29] Ritter, G.X., Wilson, J.N., *Handbook of Computer Vision Algorithms in Image Algebra*. Boca Raton, FL, CRC Press, 1996.

[30] Russell, S., Norvig, P., *Artificial Intelligence, a Modern Approach*. Prentice Hall International, 1995.

[31] Schraft, R.-D., Schmierer, G., *Service Roboter. Natick, MA*, A.K. Peters, Ltd, 2000. [also available in German from Springer-Verlag].

[32] Sciavicco, L., Siciliano, B., *Modeling and Control of Robot Manipulators*. New York, McGraw-Hill, 1996.

[33] Todd, D.J, *Walking Machines, an Introduction to Legged Robots*. Kogan Page Ltd, 1985.

Papers

[34] Aho, A.V., "Algorithms for Finding Patterns in Strings," in J. van Leeuwen (editor), *Handbook of Theoretical Computer Science*, Cambridge, MA, MIT Press, 1990, Volume A, chapter 5, 255–300.

[35] Arras, K.O., Castellanos, J.A., Siegwart, R., "Feature-Based Multi-Hypothesis Localization and Tracking for Mobile Robots Using Geometric Constraints," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'2002)*, Washington, DC, May 11–15, 2002.

[36] Arras,K.O., Persson, J., Tomatis, N., Siegwart, R., "Real-Time Obstacle Avoidance for Polygonal Robots with a Reduced Dynamic Window," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2002),* Washington, DC, May 11–15, 2002.

[37] Arras, K.O., Siegwart, R.Y., "Feature Extraction and Scene Interpretation for Map-Based Navigation and Map Building," in *Proceedings of SPIE, Mobile Robotics XII,* Vol. 3210, 1997, 42–53,.

[38] Arras, K.O., Tomatis, N., "Improving Robustness and Precision in Mobile Robot Localization by Using Laser Range Finding and Monocular Vision," in *Proceedings of the Third European Workshop on Advanced Mobile Robots (Eurobot 99)*, Zurich, September 6–9, 1999.

[39] Astolfi, A., "Exponential Stabilization of a Mobile Robot," *in Proceedings of 3rd European Control Conference,* Rome, September 1995.

[40] Barnard, K., Cardei V., Funt, B., "A Comparison of Computational Color Constancy Algorithms." *IEEE Transactions in Image Processing*. 11: 972–984, 2002.

[41] Barron, J.L., Fleet, D.J., Beauchemin, S.S., "Performance of Optical Flow Techniques." *International Journal of Computer Vision*, 12:43–77, 1994.

[42] Batavia, P., Nourbakhsh, I., "Path Planning for the Cye Robot," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00)*, Takamatsu, Japan, November, 2000.

[43] Borenstein, J., Koren, Y., "The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots." *IEEE Journal of Robotics and Automation*, 7, 278–288, 1991.

[44] Brock, O., Khatib, O., "High-Speed Navigation Using the Global Dynamic Window Approach," *in Proceeding of the IEEE International Conference on Robotics and Automation*, Detroit, May 1999.

[45] Brooks, R., "A Robust Layered Control System for a Mobile Robot," *IEEE Transactions of Robotics and Automation*, RA-2:14–23, March 1986.

[46] Brown, H.B., Zeglin, G.Z., "The Bow Leg Hopping Robot", *in Proceedings of the IEEE International Conference on Robotics and Automation*, Leuwen, Belgium, May 1998.

[47] Bruce, J., Balch,T., and Veloso, M., "Fast and Inexpensive Color Image Segmentation for Interactive Robots," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00)*, Takamatsu, Japan, October 31–November 5, 2000.

[48] Burgard,W., Cremers, A., Fox, D., Hahnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S., "Experiences with an Interactive Museum Tour-Guide Robot," *Artificial Intelligence*, 114, 1–53, 2000.

[49] Burgard, W., Derr, A., Fox, D., Cremers, A., "Integrating Global Position Estimation and Position Tracking for Mobile Robots: The Dynamic Markov Localization Approach," *in Proceedings of the 1998 IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS'98)*, Victoria, B.C., Canada, October 1998.

[50] Burgard, W., Fox, D., Henning, D., "Fast Grid-Based Position Tracking for Mobile Robots," *in Proceedings of the 21th German Conference on Artificial Intelligence (KI97)*, Freiburg, Germany, Springer-Verlag, 1997.

[51] Burgard, W., Fox, D., Jans, H., Matenar, C., Thrun, S., "Sonar-Based Mapping of Large-Scale Mobile Robot Environments using EM," *in Proceedings of the International Conference on Machine Learning*, Bled, Slovenia, 1999.

[52] Campion, G., Bastin, G., D'Andréa-Novel, B., "Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots." *IEEE Transactions on Robotics and Automation*, 12, No. 1, 47–62, 1996.

[53] Canudas de Wit, C., Sordalen, O.J., "Exponential Stabilization of Mobile Robots with Nonholonomic Constraints." *IEEE Transactions on Robotics and Automation*, 37, 1791–1797, 1993.

[54] Caprari, G., Estier, T., Siegwart, R., "Fascination of Down Scaling–Alice the Sugar Cube Robot." *Journal of Micro-Mechatronics*, 1, 177–189, 2002.

[55] Castellanos, J.A., Tardos, J.D., Schmidt, G., "Building a Global Map of the Environment of a Mobile Robot: The Importance of Correlations," *in Proceedings of the* 1997 *IEEE Conference on Robotics and Automation*, Albuquerque, NM, April 1997.

[56] Chen, C.T., Quinn, R.D., "A Crash Avoidance System Based upon the Cockroach Escape Response Circuit," *in Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.

[57] Chenavier, F., Crowley, J.L., "Position Estimation for a Mobile Robot Using Vision and Odometry," *in Proceedings of the IEEE International Conference on Robotics and Automation*, Nice, France, May 1992.

[58] Chong, K.S., Kleeman, L., "Accurate Odometry and Error Modelling for a Mobile Robot," *in Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.

[59] Choset, H., Walker, S., Eiamsa-Ard, K., Burdick, J., "Sensor-Based Exploration: Incremental Construction of the Hierarchical Generalized Voronoi Graph." *The International Journal of Robotics Research,* 19, 126–148, 2000.

[60] Cox, I.J., Leonard, J.J., "Modeling a Dynamic Environment Using a Bayesian Multiple Hypothesis Approach." *Artificial Intelligence*, 66, 311–44, 1994.

[61] Dowlingn, K., Guzikowski, R., Ladd, J., Pangels, H., Singh, S., Whittaker, W.L., "NAVLAB: An Autonomous Navigation Testbed." *Technical report CMU-RI-TR-87-24, Robotics Institute*, Pittsburgh, Carnegie Mellon University, November 1987.

[62] Dugan, B., "Vagabond: A Demonstration of Autonomous, Robust Outdoor Navigation," *in Video Proceedings of the IEEE International Conference on Robotics and Automation*, Atlanta, GA, May 1993.

[63] Elfes, A., "Sonar-Based Real World Mapping and Navigation," *in* [10].

[64] Ens, J., Lawrence, P., "An Investigation of Methods for Determining Depth from Focus." *IEEE Transactions. on Pattern Analysis and Machine Intelligence*, 15: 97–108, 1993.

[65] Espenschied, K. S., Quinn, R. D., "Biologically-Inspired Hexapod Robot Design and Simulation," *in AIAA Conference on Intelligent Robots in Field, Factory, Service and Space*, Houston, Texas, March 20–24, 1994.

[66] Falcone, E., Gockley, R., Porter, E., Nourbakhsh, I., "The Personal Rover Project: The Comprehensive Design of a Domestic Personal Robot," *Robotics and Autonomous Systems, Special Issue on Socially Interactive Robots,* 42, 245–258, 2003.

[67] Feder, H.J.S., Slotin, J-J.E., "Real-Time Path Planning Using Harmonic Potentials in Dynamic Environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.

[68] Fox, D., "KLD-Sampling: Adaptive Particle Filters and Mobile Robot Localization." *Advances in Neural Information Processing Systems 14.* MIT Press, 2001.

[69] Fox, D., Burgard,W., Thrun, S., "The Dynamic Window Approach to Collision Avoidance." *IEEE Robotics and Automation Magazine*, 4:23–33, 1997.

[70] Gander,W., Golub, G.H., Strebel, R., "Least-Squares Fitting of Circles and Ellipses." *BIT Numerical Mathematics*, vol. 34, no. 4, pp. 558–578, December 1994.

[71] Genesereth, M.R. "Deliberate Agents." *Technical Report Logic-87-2.* Stanford, CA, Stanford University, Logic Group, 1987.

[72] Golub, G., Kahan,W., "Calculating the Singular Values and Pseudo-Inverse of a Matrix." *Journal SIAM Numerical Analysis*, 2:205–223, 1965.

[73] Gutmann, J.S., Burgard, W., Fox, D., Konolige, K., "An Experimental Comparison of Localization Methods," *in Proceedings of the 1998 IEEE/RSJ International. Conference of Intelligent Robots and Systems* (IROS'98), Victoria, B.C., Canada, October 1998.

[74] Guttman, J.S., Konolige, K., "Incremental Mapping of Large Cyclic Environments," *in Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, Monterey, November 1999.

[75] Hashimoto, S., "Humanoid Robots in Waseda University - Hadaly-2 and WABIAN," *in IARP First International Workshop on Humanoid and Human Friendly Robotics*, Tsukuba, Japan, October 1998.

[76] Heale, A., Kleeman, L.: "A Real Time DSP Sonar Echo Processor," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00)*, Takamatsu, Japan, October 31–November 5, 2000.

[77] Horn, B.K.P., Schunck, B.G., "Determining Optical Flow," *Artificial Intelligence*, 17:185–203, 1981.

[78] Horswill, I., "Visual Collision Avoidance by Segmentation," *in Proceedings of IEEE International Conference on Robotics and Automation*, 902–909, 1995, IEEE Press, Munich, November 1994.

[79] Jacobs, R. and Canny, J., "Planning Smooth Paths for Mobile Robots," *in Proceeding. of the IEEE Conference on Robotics and Automation*, IEEE Press,1989, pp. 2–7.

[80] Jennings, J., Kirkwood-Watts, C., Tanis, C., "Distributed Map-making and Navigation in Dynamic Environments," *in Proceedings of the 1998 IEEE/RSJ Intl. Conference of Intelligent Robots and Systems (IROS'98)*, Victoria, B.C., Canada, October 1998.

[81] Jensfelt, P., Austin, D., Wijk, O., Andersson, M., "Feature Based Condensation for Mobile Robot Localization," *in Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, May 24–28, 2000.

[82] Kamon, I., Rivlin, E., Rimon, E., "A New Range-Sensor Based Globally Convergent Navigation Algorithm for Mobile Robots," *in Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, April 1996.

[83] Kelly, A., "Pose Determination and Tracking in Image Mosaic Based Vehicle Position Estimation," *in Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00)*, Takamatsu, Japan, October 31–November 5, 2000.

[84] Khatib, M., Chatila, R., "An Extended Potential Field Approach for Mobile Robot Sensor-Based Motions," *in Proceedings of the Intelligent Autonomous Systems IAS-4*, IOS Press, Karlsruhe, Germany, March 1995, pp. 490–496.

[85] Khatib, M., Jaouni, H., Chatila, R., Laumod, J.P., "Dynamic Path Modification for Car-Like Nonholonomic Mobile Robots," *in Proceedings of IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.

[86] Khatib, O., Quinlan, S., "Elastic Bands: Connecting, Path Planning and Control," *in Proceedings of IEEE International Conference on Robotics and Automation*, Atlanta, GA, May 1993.

[87] Ko, N.Y., Simmons, R., "The Lane-Curvature Method for Local Obstacle Avoidance," *in Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98)*, Victoria, B.C., Canada, October 1998.

[88] Koenig, S., Simmons, R., "Xavier: A Robot Navigation Architecture Based on Partially Observable Markov Decision Process Models," *in* [20]

[89] Konolige, K.,. "A Gradient Method for Realtime Robot Control," *in Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, Takamatsu, Japan, 2000.

[90] Konolige, K., "Small Vision Systems: Hardware and Implementation," in P*roceedings of Eighth International Symposium on Robotics Research*, Hayama, Japan, October 1997.

[91] Koperski, K., Adhikary, J., Han, J., "Spatial Data Mining: Progress and Challenges Survey Paper,". *in Proceedings of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, Montreal, June 1996.

[92] Koren, Y., Borenstein, J., "High-Speed Obstacle Avoidance for Mobile Robotics," *in Proceedings of the IEEE Symposium on Intelligent Control*, Arlington, VA, August 1988, pp. 382-384.

[93] Koren, Y., Borenstein, J., "Real-Time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments," *in Proceedings of the IEEE International Conference on Robotics and Automation*, Los Alamitos, CA, May 1990.

[94] Kuipers, B., Byun, Y.-T., "A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations." *Journal of Robotics and Autonomous Systems*, 8:47–63, 1991.

[95] Lamon, P., Nourbakhsh, I., Jensen, B., Siegwar,t R., "Deriving and Matching Image Fingerprint Sequences for Mobile Robot Localization," *in Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 21–26, 2001.

[96] Latombe, J-C., Barraquand, J., "Robot Motion Planning: A Distributed Presentation Approach." *International Journal of Robotics Research*, 10: 628–649, 1991.

[97] Lauria, M., Estier, T., Siegwart, R.: "An Innovative Space Rover with Extended Climbing Abilities." *in Video Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, May 24–28, 2000.

[98] Lazanas, A., Latombe, J.-C., "Landmark-Based Robot Navigation," *in Proceedings of the Tenth National Conference on AI*. San Jose, CA, July 1992.

[99] Lazanas, A. Latombe, J.C., "Motion Planning with Uncertainty: A Landmark Approach." *Artificial Intelligence*, 76:285–317, 1995.

[100] Lee, S.-O., Cho, Y.-J., Hwang-Bo, M., You, B.-J., Oh, S.-R.: "A Stabile Target-Tracking Control for Unicycle Mobile Robots," *in Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Takamatsu, Japan, October 31–November 5, 2000.

[101] Lumelsky, V., Skewis, T., "Incorporating Range Sensing in the Robot Navigation Function." *IEEE Transactions on Systems, Man, and Cybernetics*, 20:1990, pp. 1058–1068.

[102] Lumelsky, V., Stepanov, A., "Path-Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape," in [10].

[103] Maes, P., "The Dynamics of Action Selection," *in Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, 1989, pp. 991–997.

[104] Maes, P., "Situated Agents Can Have Goals," *Robotics and Autonomous Systems*, 6: 49–70. 1990.

[105] Martinelli, A., Siegwart, R., "Estimating the Odometry Error of a Mobile Robot during Navigation," *in Proceedings of the European Conference on Mobile Robots (ECMR 2003)*, Warsaw, September 4–6, 2003.

[106] Maybeck,P.S., "The Kalman Filter: An Introduction to Concepts," *in* [10].

[107] Minguez, J., Montano, L., "Nearness Diagram Navigation (ND): A New Real Time Collision Avoidance Approach," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Takamatsu, Japan, October 2000.

[108] Minguez, J., Montano, L., "Robot Navigation in Very Complex, Dense, and Cluttered Indoor / Outdoor Environments," *in Proceeding of International Federation of Automatic Control (IFAC2002)*, Barcelona, April 2002.

[109] Minguez, J., Montano, L., Khatib, O., "Reactive Collision Avoidance for Navigation with Dynamic Constraints," *in Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

[110] Minguez, J., Montano, L., Simeon, T., Alami, R., "Global Nearness Diagram Navigation (GND)," *in Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, 2001.

[111] Montano, L., Asensio, J.R., "Real-Time Robot Navigation in Unstructured Environments Using a 3D Laser Range Finder," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, IROS 97, IEEE Press, Vol. 2. Grenoble, France, September 1997, pp. 526–532.

[112] Moravec, H. and Elfes, A.E., "High Resolution Maps from Wide Angle Sonar," *in Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, IEEE Press, At.-Louis, MO, March 1985, pp. 116–121.

[113] Moutarlier, P., Chatila, R., "Stochastic Multisensory Data Fusion for Mobile Robot Location and Environment Modelling," *in Proceedings of the 5th International Symposium of Robotics Research*, Tokyo, 1989, pp. 207–216.

[114] Nayar, S.K., "Catadioptric Omnidirectional Camera." *IEEE CVPR*, pp. 482–488, 1997.

[115] Nilsson, N.J., "Shakey the Robot." *SRI, International, Technical Note*, Menlo Park, CA, 1984, No. 325.

[116] Nourbakhsh, I.R., "Dervish: An Office-Navigation Robot," *in* [20].

[117] Nourbakhsh, I.R., Andre. D., Tomasi, C., Genesereth, M.R., "Mobile Robot Obstacle Avoidance via Depth from Focus," *Robotics and Autonomous Systems*, 22:151–158, 1997.

[118] Nourbakhsh, I.R., Bobenage, J., Grange, S., Lutz, R., Meyer, R, Soto, A., "An Affective Mobile Educator with a Full-Time Job." *Artificial Intelligence*, 114:95–124, 1999.

[119] Nourbakhsh, I.R., Powers, R., Birchfield, S., "DERVISH, an Office-Navigation Robot." *AI Magazine*, 16:39–51, summer 1995.

[120] Pell, B., Bernard, D., Chien, S., Gat, E., Muscettola, N., Nayak, P., Wagner, M., Williams, B., "An Autonomous Spacecraft Agent Prototype." *Autonomous Robots*, No. 5, 1–27, 1998.

[121] Pentland, A.P., "A New Sense for Depth of Field." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 9:523–531, 1987.

[122] Philippsen, R., Siegwart, R., "Smooth and Efficient Obstacle Avoidance for a Tour Guide Robot," *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2003)*, Taipei, Taiwan, 2003.

[123] Pratt, J., Pratt, G., "Intuitive Control of a Planar Bipedal Walking Robot," *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '98)*, Leuven, Belgium, May 16–21, 1998

[124] Raibert, M. H., Brown, H. B., Jr., Chepponis, M., "Experiments in balance with a 3D One-Legged Hopping Machine." *International Journal of Robotics Research*, 3:75–92, 1984.

[125] Ringrose, R., "Self-Stabilizing Running," *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '97)*, Albuquerque, NM, April 1997.

[126] Rowe, A., Rosenberg, C., Nourbakhsh, I., "A Simple Low Cost Color Vision System," *in Proceedings of Tech Sketches for CVPR 2001*. Kuaii, Hawaii, December 2001.

[127] Rubner, Y., Tomasi, C., Guibas, L., "The Earth Mover's Distance as a Metric for Image Retrieval," *STAN-CS-TN-98-86, Stanford University*, 1998.

[128] Schlegel, C., "Fast Local Obstacle under Kinematic and Dynamic Constraints," *in Proceedings of the IEEE International Conference on Intelligent Robot and Systems (IROS 98)*, Victoria, B.C. Canada 1998, pp. 594–599.

[129] Schultz, A., Adams, W., "Continuous Localization Using Evidence Grids," *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '98)*, May 16–21, 1998, Leuven, Belgium, pp.2833–2839 [also available as NCARAI Report AIC-96-007].

[130] Schweitzer, G., Werder, M., "ROBOTRAC – a Mobile Manipulator Platform for Rough Terrain," *in Proceedings of the International Symposium on Advanced Robot Technology (ISART)*, Tokyo, Japan, March, 1991.

[131] Shi, J., Malik, J., "Normalized Cuts and Image Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 82:888–905, 2000.

[132] Siegwart R., et al., (2003) "Robox at Expo.02: A Large Scale Installation of Personal Robots." *Journal of Robotics and Autonomous Systems*, 42:203–222, 2003.

[133] Siegwart, R., Lamon, P., Estier, T., Lauria, M, Piguet, R., "Innovative Design for Wheeled Locomotion in Rough Terrain," *Journal of Robotics and Autonomous Systems*, 40:151–162, 2002.

[134] Simhon, S., Dudek, G., "A Global Topological Map Formed by Local Metric Maps," *in Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98)*, Victoria, B.C., Canada, October 1998.

[135] Simmons, R., "The Curvature Velocity Method for Local Obstacle Avoidance," *in Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, April 1996.

[136] Smith, R., Self, M., Cheeseman, P., "Estimating Uncertain Spatial Relationships in Robotics," *Autonomous Robot Vehicles*, I. J. Cox and G. T. Wilfong (editors), Springer-Verlag, 1990, pp. 167–193.

[137] Sordalen, O.J., Canudas de Wi,t C., "Exponential Control Law for a Mobile Robot: Extension to Path Following," *IEEE Transactions on Robotics and Automation*, 9:837–842, 1993.

[138] Steinmetz, B.M., Arbter, K., Brunner, B., Landzettel, K., "Autonomous Vision-Based Navigation of the Nanokhod Rover," *in Proceedings of i-SAIRAS 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Montreal, June 18–22, 2001.

[139] Stentz, A., "The Focussed D* Algorithm for Real-Time Replanning," in *Proceedings of IJCAI-95*, August 1995.

[140] Stevens, B.S., Clavel, R., Rey, L., "The DELTA Parallel Structured Robot, Yet More Performant through Direct Drive," *in Proceedings of the 23rd International Symposium on Industrial Robots*, Barcelona, October 1992, pp. 485–493.

[141] Takeda, H., Facchinetti, C., Latombe, J.C., "Planning the Motions of a Mobile Robot in a Sensory Uncertainty Field." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:1002–1017, 1994.

[142] Thrun, S., Burgard, W., Fox, D., "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots." *Autonomous Robots*, 31:1–25. 1998.

[143] Thrun, S., et al., "Monrovia: A Second Generation Museum Tour-Guide Robot," *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'99)*, Detroit, May 1999.

[144] Thrun, S., Fox, D., Burgard, W. Dellaert, F., "Robust Monte Carlo Localization for Mobile Robots," *Artificial Intelligence*, 128:99–141, 2001.

[145] Thrun, S., Gutmann, J.-S., Fox, D., Burgard, W., Kuipers, B., "Integrating Topological and Metric Maps for Mobile Robot Navigation: A Statistical Approach," *in Proceedings of the National Conference on Artificial Intelligence (AAAI)*,1998.

[146] Tomasi, C., Shi, J., "Image Deformations Are Better Than Optical Flow." *Mathematical and Computer Modelling*, 24:165–175, 1996.

[147] Tomatis, N., Nourbakhsh, I., Siegwart, R., "Hybrid Simultaneous Localization and Map Building: A Natural Integration of Topological and Metric." *Robotics and Autonomous Systems* 44, 3–14, 2003.

[148] Tzafestas, C.S., Tzafestas, S.G., "Recent Algorithms for Fuzzy and Neurofuzzy Path Planning and Navigation of Autonomous Mobile Robots," *Systems-Science*, 25:25–39, 1999.

[149] Ulrich, I., Borenstein, J., "VFH*: Local Obstacle Avoidance with Look-Ahead Verification," *in Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, May 24–28, 2000.

[150] Ulrich, I., Borenstein, J., "VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots," *in Proceedings of the International Conference on Robotics and Automation (ICRA'98)*, Leuven, Belgium, May 1998.

[151] Ulrich, I., Nourbakhsh, I., "Appearance-Based Obstacle Detection with Monocular Color Vision," *in the Proceedings of the AAAI National Conference on Artificial Intelligence*. Austin, TX. August 2000.

[152] Ulrich, I., Nourbakhsh, I., "Appearance-Based Place Recognition for Topological Localization," *in Proceedings of t he IEEE International Conference on Robotics and Automation*, San Francisco, pp. 1023–1029, April 2000.

[153] Vanualailai, J., Nakagiri, S., Ha, J-H., "Collision Avoidance in a Two-Point System via Liapunov's Second Method." *Mathematics and Simulation*, 39:125–141, 1995.

[154] Van Winnendael, M., Visenti G., Bertrand, R., Rieder, R., "Nanokhod Microrover Heading towards Mars," *in Proceedings of the Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space* (ESA SP-440), Noordwijk, Netherlands, pp. 69–76, 1999.

[155] Weiss, G., Wetzler, C., Puttkamer, E., "Keeping Track of Position and Orientation of Moving Indoor Systems by Correlation of Range-Finder Scans," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, Munich, pp.595–601, September 12-16, 1994.

[156] Wullschleger, F.H., Arra,s K.O., Vestli, S.J., "A Flexible Exploration Framework for Map Building," *in Proceedings of the Third European Workshop on Advanced Mobile Robots (Eurobot 99)*, Zurich, September 6-9, 1999.

[157] Yamauchi, B., Schultz, A., Adams, W., "Mobile Robot Exploration and Map-Building with Continuous Localization," *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '98)*, Leuven, Belgium, May 1998.

[158] Zhang, Z., "A Flexible New Technique for Camera Calibration." *Microsoft Research Technical Report 98-71.*, December 1998
[see also http://research.microsoft.com/~zhang].

Referenced Webpages

[159] Fisher, R.B. (editor), "CVonline: On-line Compendium of Computer Vision," Available at www.dai.ed.ac.uk/CVonline/.

[160] The Intel Image Processing Library: http://developer.intel.com/software/products/perflib/ipl/.

[161] Source code release site: www.cs.cmu.edu/~jbruce/cmvision.

[162] Newton Labs website: www.newtonlabs.com.

[163] For probotics: http://www.personalrobots.com.

Interesting Internet Links to Mobile Robots

General homepages with mainly mobile robots

http://ranier.hq.nasa.gov/telerobotics_page/coolrobots.html
http://www-robotics.cs.umass.edu/robotics.html
http://www.ai.mit.edu/projects/mobile-robots/
http://www.ri.cmu.edu/project_lists/index.html
http://www.roboticsclub.org/links.html
http://www.activrobots.com/
http://asl.epfl.ch (at EPFL)
http://robotics.epfl.ch (at EPFL)
http://www.cs.cmu.edu/~illah/EDUTOY (at CMU)
http://www.cs.cmu.edu/~mercator/ (at CMU)

Homepages with mainly wheeled mobile robots

http://www.laas.fr/RIA/RIA.html
http://www.cs.cmu.edu/~illah/lab.html
http://www.cs.umd.edu/projects/amrl/amrl.html
http://www.cc.gatech.edu/ai/robot-lab/
http://www.engin.umich.edu/research/mrl/index.html

Homepages with walking and climbing robots

http://www.uwe.ac.uk/clawar/
http://www.fzi.de/divisions/ipt/
http://www.automation.hut.fi/
http://www.ai.mit.edu/projects/leglab/
http://asl.epfl.ch/
http://www.cs.cmu.edu/~personalrover/

Homepages for robots on the Web

http://telerobot.mech.uwa.edu.au/secn/links.html
http://queue.IEOR.Berkeley.EDU/~goldberg/art/telerobotics-links.html
http://www.cs.uni-bonn.de/~rhino/
http://www.cs.cmu.edu/~minerva/
http://telerobot.mech.uwa.edu.au
http://www.ieor.berkeley.edu/~goldberg/GC/www.html
http://pumapaint.rwu.edu/
http://mars.graham.com/wits/
http://www.cs.cmu.edu/~illah/SAGE/index.html
http://www.dislocation.net/
http://rr-vs.informatik.uni-ulm.de/rr/
http://www.eventscope.org/Eventscope/main.htm