基于ROS的移动跟随机器人

**第一章 绪论**

**1.1 引言**

21世纪是什么样的世纪？是物联网的世纪？是VR的世纪？也许吧，但我更相信21世纪是机器人的世纪。目前，我国已经步入经济转型的拐点区间，人口红利越来越难以支撑中国经济的发展和进步。在很多行业都已经开始了机器换人，生产工艺升级换代的步伐。工信部，发改委，财政部日前联合印发了《机器人产业规划（2016-2020）》。由此可见，机器人未来的政策空间和试场发展空间都是非常巨大的。一方面，发展工业机器人在满足我国制造业的转型升级，提质增效，实现“中国制造2025”等方面具有极为重大的意义，是全面推进实施制造强国战略的重要一步。另一方面，从服务机器人来说，也要满足未来市场需求的增长。首先，这包括了基本生活需求，比如养老，助老，助残等。其次是国家安全需要，比如救灾，抢险，海底勘探，航天，国防。最后还有家庭服务和娱乐机器人，比如娱乐，儿童教育，智能家居应用等。

智能移动机器人是机器人研究的一个重要分支。它需要具备强健的“肢”，明亮的“眼”，灵巧的“嘴”以及聪慧的“脑”，这一切的实现实际上涉及诸多技术领域。目前移动机器人涉及的主要技术有运动控制，环境感知，导航与定位，最优路径规划等。

本文主要介绍了视觉目标跟踪和路径规划技术，利用Kinect摄像头构建了可在较为复杂的环境中运行的智能移动机器人系统。

**1.2 移动机器人发展史**

至今，在工业制造领域，机器人学已经取得了巨大的成功。机械臂，机械灵巧手，焊接机器人以及喷漆机器人等构成了巨大的工业产值。但是，对于所有的这些成功应用，都有一个共同的缺点：缺乏机动性。固定的机械手被安装在固定的地方，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才能。

60年代后期，美国和苏联为完成月球探测计划，研制并应用了移动机器人。美国“探测者”3号,其操作器在地面的遥控下，完成了在月球上挖沟和执行其他任务。苏联的“登月者”20号在无人驾驶的情况下降落在月球表面，操作器在月球表面钻削岩石，并把土壤和岩石样品装进回收容器并送回地球。70年代初期，日本早稻田大学研制出具有仿人功能的两足步行机器人。为适应原子能利用和海洋开发的需要，极限作业机器人和水下机器人也发展较快。国内的机器人发展较晚，但近年来在政府的支持下，也有了长足的进步。

**1.3 ROS（Robot Operating System）简介**

ROS是一个分布式的系统，主要的的构成是节点（node）,每一个节点可以就看做一个单独的独立单元，它可以接收外部的消息（这个步骤叫订阅，subcribe）,也可以向外部发出自己的消息（这个步骤叫发布，publish）。服务与消息机制是节点功能实现的一个典型的方式，也就是接收外部的消息，并完成对该消息的响应。这里有个经典的应用例子:一个提供加法运算服务的节点，定时查询消息，一旦接收到消息（俩个数值），立即返回加法运算的结果（俩数之和），节点是一个单独的处理单元，这样会大大简化程序设计和代码量，每个节点在代码层看起来就是一个独立的执行文件，有一个main()函数入口。

除此之外，节点之间的通讯是通过类似TCP通讯机制的通讯底层所实现的，所以整个ROS上的节点是否跑在同一运算平台上并不重要（不考虑运算能力），只要这些节点之间有高效的通讯通道，便可以让节点相对运算平台透明，这也是分布式系统的特点。还有一系列的仿真，调试工具，如gazebo,rviz,rqt等，更重要的是能够使用多种语言进行开发，如C++,python等。

**1.4 视觉目标跟踪技术**

目标跟踪是绝大多数视觉系统中不可或缺的环节。在二维视频跟踪算法中，基于目标颜色信息或基于目标运动信息等方法是常用的跟踪方法。从以往的研究中我们发现，大多数普通摄像头（彩色摄像头）下非基于背景建模的跟踪算法都极易受光照条件的影响。这是因为颜色变化在某种程度上是光学的色彩变化造成的。如基于体素和图像像素守恒假设的光流算法它也是假设一个物体的颜色在前后两帧没有巨大而明显的变化。

目标跟踪的实质实际上是模板匹配的问题，它的思想很简单，我们把要跟踪的目标保存好，然后在每一帧来临的时候，我们在整个图像中寻找与这个目标最相似的，我们就相信这个就是目标了。21世纪以来，人们以现代滤波方法的概率形式来判断目标的匹配程度，常见的有卡尔曼滤波，扩展卡尔曼滤波，粒子滤波，相关滤波等。但模板匹配的前提是目标特征的选取，人们通常使用的图像中的目标特征有颜色特征，边缘特征以及光流特征等，第3章我们还会详细的描述。

**1.5 本文主要研究内容**

本文结合跟随机器人的主要技术难点，对目标跟踪的相关知识进行了比较深入的研究。

我们采用微软的kinect摄像机作为移动机器人的主要环境信息感知器，对kinect摄像机的结构，性能参数做了介绍，分析了其作为移动机器人传感器的一些优点和缺点。

为了实现视觉目标的跟踪，我们探讨了基于核相关滤波的目标跟踪算法，并且最终将KCF算法集成到ROS系统中，实现了对跟随系统的设计。

**第二章 机器人硬件系统**

**2.1 整体框架**

为了能够实现目标跟踪与路径规划，我们采用微软公司的kinect作为移动机器人的“眼睛”，用来获取跟踪目标与所在环境的信息。移动机器人采用M-robot,是一群机器人爱好者开发的一款简易移动机器人。

系统如图2.1所示。



图2.1 硬件平台

**2.2 M-robot机器人介绍**

这个平台基于ROS开源系统构建，可实现室内建立地图和导航，能够作为ROS的学习平台，不能达到消费级产品的稳定性导航效果，可以在开源算法的基础上继续优化得到更好地效果。该移动机器人系统包括：

1. 底层采用自制的Arduino2560控制（2560+328双控制芯片），遵循iRobot开源协议。
2. 拥有独立于ROS之外的接口：4个超声波接口，2个舵机控制接口，1个UART接口，一个I2C接口。
3. 上位机采用笔记本电脑。
4. 系统参数：净重3KG，负载能力5KG，主动轮2个，全向轮1个，采用差分驱动方式，精度可达厘米级别，速度为0~1.0m/s,电源输入为DC-12V，通信波特率为57600/115200。

2.2.1 主控制器

主要资源如图2.2所示：

（1）主控芯片： Atmega 2560

（2）超声波控制芯片：Atmega328

（3）USB--TTL：FT232

（4）电源芯片：LM2596--5V

（5）板载接口：电源输入接口，5V电压输出接口，开关，USB串口，编码器输入接口，电机驱动控制接口，超声波接口，舵机接口，遥控接口，下载固件接口以及预留接口。

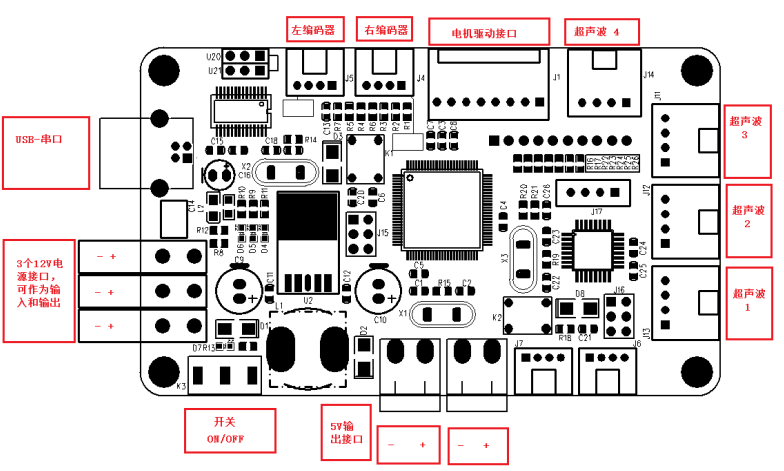


图2.2

2.2.2 电机驱动板介绍

电机驱动板适用于直流电机驱动，理论电流100A，如果要控制大功率电机，需要加散热器。实物图如下：

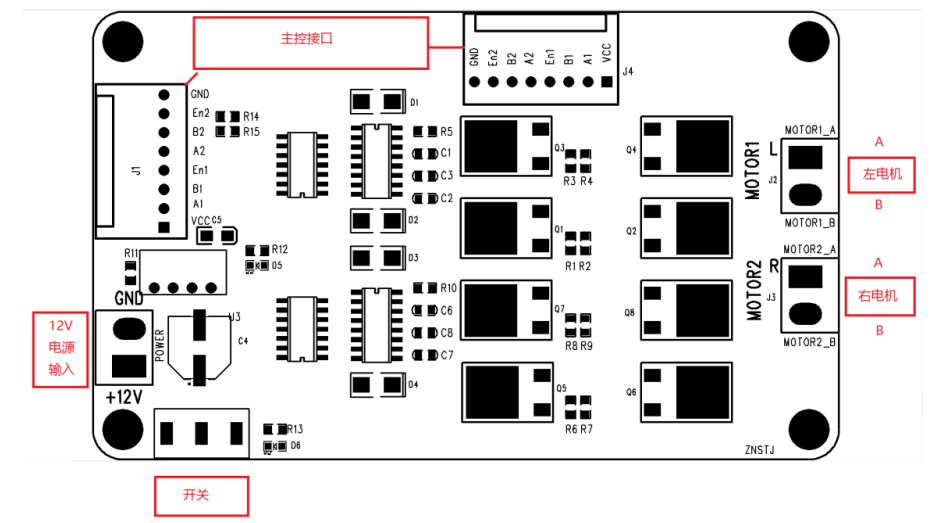


图2.3

板载接口有：

1. 电源输入接口，用于电机电源7--12V。
2. 电机控制接口。
3. 总电源开关。

2.2.3 电源板介绍

适用于锂电池和蓄电池，实物如下图2.4所示：

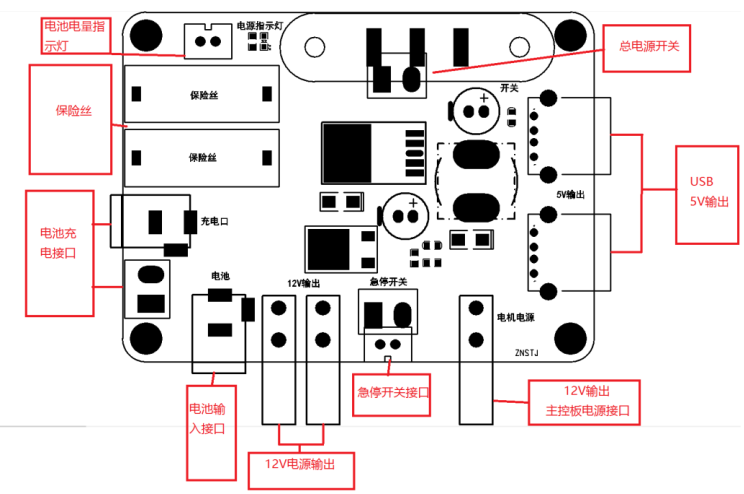


图2.4

板载资源有:

1. 电池输入接口，DC-5V
2. 电池充电口，DC-5V
3. 俩个10A保险丝
4. 电量指示模块接口
5. 总电源开关
6. 急停开关接口
7. 12V主控接口，急停开关控制
8. 12V输出，不受急停开关控制
9. 5V输出

**2.3 Kinect**

Kinect for Xbox 360，简称 Kinect，是由微软开发，应用于Xbox 360 主机的周边设备。它让玩家不需要手持或踩踏控制器，而是使用语音指令或手势来操作 Xbox360 的系统界面。它也能捕捉玩家全身上下的动作，用身体来进行游戏，带给玩家“免控制器的游戏与娱乐体验”。其在2010年11月4日于美国上市，建议售价149美金。Kinect在销售前60天内，卖出八百万部，目前已经申请吉尼斯世界记录，成为全世界销售最快的消费性电子产品。

2012年2月1日，微软正式发布面向Windows系统的Kinect版本“Kinect for Windows”,建议售价249美金。而在2012年晚些时候，微软还将发布面向“教育用户”的特别版Kinect。

2.3.1 kinect硬件介绍

Kinect有三个镜头，中间的镜头是 RGB 彩色摄影机，用来采集彩色图像。左右两边镜头则分别为红外线发射器和红外线CMOS 摄影机所构成的3D结构光深度感应器，用来采集深度数据（场景中物体到摄像头的距离）。彩色摄像头最大支持1280\*960分辨率成像，红外摄像头最大支持640\*480成像。Kinect还搭配了追焦技术，底座马达会随着对焦物体移动跟着转动。Kinect也内建阵列式麦克风，由四个麦克风同时收音，比对后消除杂音，并通过其采集声音进行语音识别和声源定位。

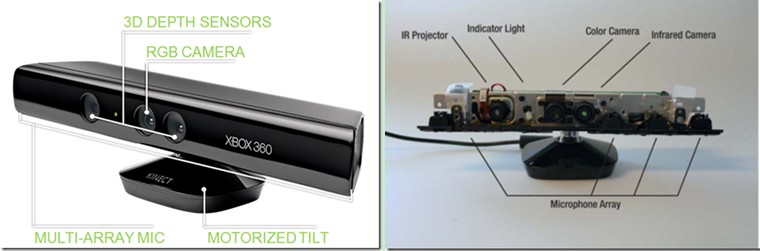


图2.5

2.3.2 kinect标定

**第三章 ROS介绍**

**3.1 ROS概览**

随着[机器人](http://lib.csdn.net/base/robot" \o "机器人知识库" \t "http://blog.csdn.net/pinbodexiaozhu/article/details/_blank)领域的快速发展和复杂化，代码的复用性和模块化的需求原来越强烈，而已有的开源机器人系统又不能很好的适应需求。2010年Willow Garage公司发布了开源机器人[操作系统](http://lib.csdn.net/base/operatingsystem" \o "操作系统知识库" \t "http://blog.csdn.net/pinbodexiaozhu/article/details/_blank)ROS（robot operating system），很快在机器人研究领域展开了学习和使用ROS的热潮。

  ROS系统是起源于2007年斯坦福大学人工[智能](http://lib.csdn.net/base/aiplanning" \o "人工智能规划与决策知识库" \t "http://blog.csdn.net/pinbodexiaozhu/article/details/_blank)实验室的项目与机器人技术公司Willow Garage的个人机器人项目（Personal Robots Program）之间的合作，2008年之后就由Willow Garage来进行推动。已经有四年多的时间了 。随着PR2那些不可思议的表现，譬如叠衣服，插插座，做早饭，ROS也得到越来越多的关注。Willow Garage公司也表示希望借助开源的力量使PR2变成“全能”机器人。

  PR2价格高昂，2011年零售价高达40万美元。PR2现主要用于研究。PR2有两条手臂，每条手臂七个关节，手臂末端是一个可以张合的钳子。PR2依靠底部的四个轮子移动。在PR2的头部，胸部，肘部，钳子上安装有高分辨率摄像头，激光测距仪，惯性测量单元，触觉传感器等丰富的传感设备。在PR2的底部有两台8核的电脑作为机器人各硬件的控制和通讯中枢。两台电脑安装有Ubuntu和ROS。

**3.2 ROS特点**

  ROS是开源的，是用于机器人的一种后操作系统，或者说次级操作系统。它提供类似操作系统所提供的功能，包含硬件抽象描述、底层驱动程序管理、共用功能的执行、程序间的消息传递、程序发行包管理，它也提供一些工具程序和库用于获取、建立、编写和运行多机整合的程序。、

****

图3.1

  ROS的首要设计目标是在机器人研发领域提高代码复用率。ROS是一种分布式处理框架（又名Nodes）。这使可执行文件能被单独设计，并且在运行时松散耦合。这些过程可以封装到数据包（Packages）和堆栈（Stacks）中，以便于共享和分发。ROS还支持代码库的联合系统。使得协作亦能被分发。这种从文件系统级别到社区一级的设计让独立地决定发展和实施工作成为可能。上述所有功能都能由ROS的基础工具实现。

ROS的运行架构是一种使用ROS通信模块实现模块间P2P的松耦合的网络连接的处理架构，它执行若干种类型的通讯，包括基于服务的同步RPC（远程过程调用）通讯、基于Topic的异步数据流通讯，还有参数服务器上的数据存储。但是ROS本身并没有实时性。

ROS的主要特点可以归纳为以下几条：

（1）点对点设计

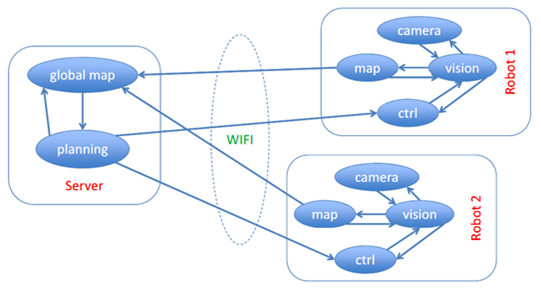


图3.2

一个使用ROS的系统包括一系列进程，这些进程存在于多个不同的主机并且在运行过程中通过端对端的拓扑结构进行联系。虽然基于中心服务器的那些软件框架也可以实现多进程和多主机的优势，但是在这些框架中，当各电脑通过不同的网络进行连接时，中心数据服务器就会发生问题。

ROS的点对点设计以及服务和节点管理器等机制可以分散由计算机视觉和语音识别等功能带来的实时计算压力，能够适应多机器人遇到的挑战。

（2）多语言支持

在写代码的时候，许多编程者会比较偏向某一些编程语言。这些偏好是个人在每种语言的编程时间、调试效果、语法、执行效率以及各种技术和文化的原因导致的结果。为了解决这些问题，我们将ROS设计成了语言中立性的框架结构。ROS现在支持许多种不同的语言，例如C++、Python、Octave和LISP，也包含其他语言的多种接口实现。



图3.3

ROS的特殊性主要体现在消息通讯层，而不是更深的层次。端对端的连接和配置利用XML-RPC机制进行实现，XML-RPC也包含了大多数主要语言的合理实现描述。我们希望ROS能够利用各种语言实现的更加自然，更符合各种语言的语法约定，而不是基于C语言给各种其他语言提供实现接口。然而，在某些情况下利用已经存在的库封装后支持更多新的语言是很方便的，比如Octave的客户端就是通过C++的封装库进行实现的。

为了支持交叉语言，ROS利用了简单的、语言无关的接口定义语言去描述模块之间的消息传送。接口定义语言使用了简短的文本去描述每条消息的结构，也允许消息的合成，例如下图就是利用接口定义语言描述的一个点的消息：

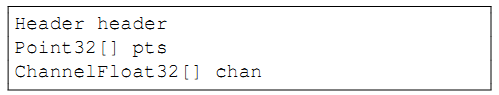


图 3.4

每种语言的代码产生器就会产生类似本种语言目标文件，在消息传递和接收的过程中通过ROS自动连续并行的实现。这就节省了重要的编程时间，也避免了错误：之前3行的接口定义文件自动的扩展成137行的C++代码，96行的Python代码，81行的Lisp代码和99行的Octave代码。因为消息是从各种简单的文本文件中自动生成的，所以很容易列举出新的消息类型。在编写的时候，已知的基于ROS的代码库包含超过四百种消息类型，这些消息从传感器传送数据，使得物体检测到了周围的环境。

最后的结果就是一种语言无关的消息处理，让多种语言可以自由的混合和匹配使用。

（3）精简与集成

大多数已经存在的机器人软件工程都包含了可以在工程外重复使用的驱动和算法，不幸的是，由于多方面的原因，大部分代码的中间层都过于混乱，以至于很困难提取出它的功能，也很难把它们从原型中提取出来应用到其他方面。

为了应对这种趋势，我们鼓励将所有的驱动和算法逐渐发展成为和ROS没有依赖性单独的库。ROS建立的系统具有模块化的特点，各模块中的代码可以单独编译，而且编译使用的CMake工具使它很容易的就实现精简的理念。ROS基本将复杂的代码封装在库里，只是创建了一些小的应用程序为ROS显示库的功能，就允许了对简单的代码超越原型进行移植和重新使用。作为一种新加入的有优势，单元测试当代码在库中分散后也变得非常的容易，一个单独的测试程序可以测试库中很多的特点。

ROS利用了很多现在已经存在的开源项目的代码，比如说从Player项目中借鉴了驱动、运动控制和仿真方面的代码，从OpenCV中借鉴了视觉算法方面的代码，从OpenRAVE借鉴了规划算法的内容，还有很多其他的项目。在每一个实例中，ROS都用来显示多种多样的配置选项以及和各软件之间进行数据通信，也同时对它们进行微小的包装和改动。ROS可以不断的从社区维护中进行升级，包括从其他的软件库、应用补丁中升级ROS的源代码。

1. 工具包丰富

为了管理复杂的ROS软件框架，我们利用了大量的小工具去编译和运行多种多样的ROS组建，从而设计成了内核，而不是构建一个庞大的开发和运行环境。

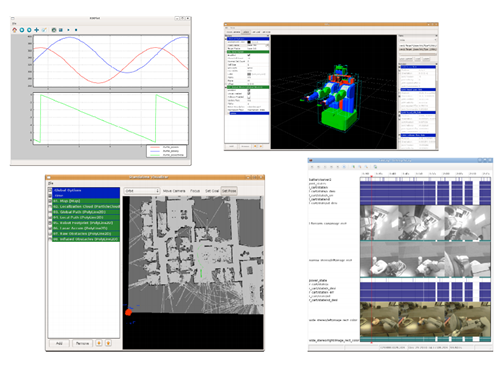


图 3.5

这些工具担任了各种各样的任务，例如，组织源代码的结构，获取和设置配置参数，形象化端对端的拓扑连接，测量频带使用宽度，生动的描绘信息数据，自动生成文档等等。尽管我们已经测试通过像全局时钟和控制器模块的记录器的核心服务，但是我们还是希望能把所有的代码模块化。我们相信在效率上的损失远远是稳定性和管理的复杂性上无法弥补的。

1. 免费且开源

ROS所有的源代码都是公开发布的。我们相信这将必定促进ROS软件各层次的调试，不断的改正错误。虽然像Microsoft Robotics Studio和Webots这样的非开源软件也有很多值得赞美的属性，但是我们认为一个开源的平台也是无可为替代的。当硬件和各层次的软件同时设计和调试的时候这一点是尤其真实的。

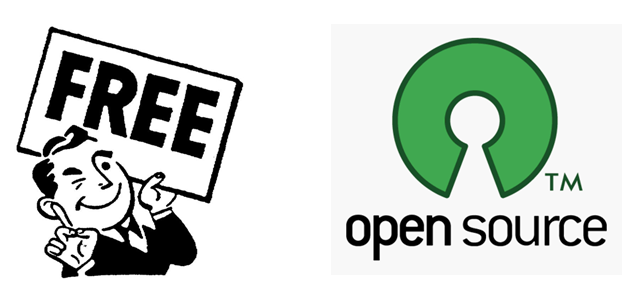


图 3.6

ROS以分布式的关系遵循这BSD许可，也就是说允许各种商业和非商业的工程进行开发。ROS通过内部处理的通讯系统进行数据的传递，不要求各模块在同样的可执行功能上连接在一起。如此，利用ROS构建的系统可以很好的使用他们丰富的组件：个别的模块可以包含被各种协议保护的软件，这些协议从GPL到BSD，但是许可的一些“污染物”将在模块的分解上就完全消灭掉。

**3.3 ROS总体框架**

根据ROS系统代码的维护者和分布来标示，主要有两大部分：

（1）main：核心部分，主要由Willow Garage公司和一些开发者设计、提供以及维护。它提供了一些分布式计算的基本工具，以及整个ROS的核心部分的程序编写。

（2）universe：全球范围的代码，有不同国家的ROS社区组织开发和维护。一种是库的代码，如OpenCV、PCL等；库的上一层是从功能角度提供的代码，如人脸识别，他们调用下层的库；最上层的代码是应用级的代码，让机器人完成某一确定的功能。

一般是从另一个角度对ROS分级的，主要分为三个级别：计算图级、文件系统级、社区级。

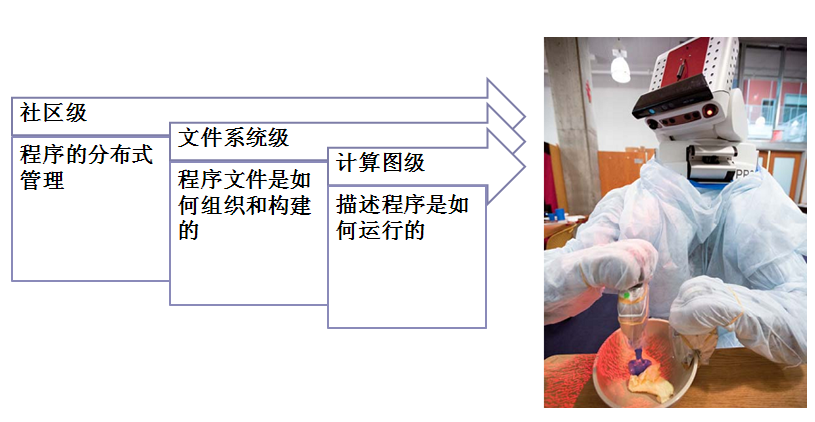


图 3.7

2.3.1 计算图级

计算图是ROS处理数据的一种点对点的网络形式。程序运行时，所有进程以及他们所进行的数据处理，将会通过一种点对点的网络形式表现出来。这一级主要包括几个重要概念：节点（node）、消息（message）、主题（topic）、服务（service）。



图3.8

（1）节点

节点就是一些直行运算任务的进程。ROS利用规模可增长的方式是代码模块化：一个系统就是典型的由很多节点组成的。在这里，节点也可以被称之为“软件模块”。我们使用“节点”使得基于ROS的系统在运行的时候更加形象化：当许多节点同时运行时，可以很方便的将端对端的通讯绘制成一个图表，在这个图表中，进程就是图中的节点，而端对端的连接关系就是其中弧线连接。

（2）消息

节点之间是通过传送消息进行通讯的。每一个消息都是一个严格的数据结构。原来标准的数据类型（整型，浮点型，布尔型等等）都是支持的，同时也支持原始数组类型。消息可以包含任意的嵌套结构和数组（很类似于C语言的结构structs）。

（3）主题

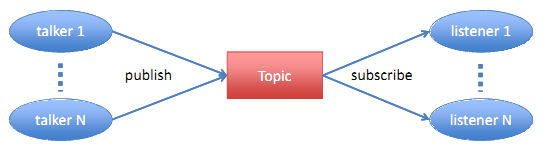
****

图3.9

消息以一种发布/订阅的方式传递。一个节点可以在一个给定的主题中发布消息。一个节点针对某个主题关注与订阅特定类型的数据。可能同时有多个节点发布或者订阅同一个主题的消息。总体上，发布者和订阅者不了解彼此的存在。

（4）服务

虽然基于话题的发布/订阅模型是很灵活的通讯模式，但是它广播式的路径规划对于可以简化节点设计的同步传输模式并不适合。在ROS中，我们称之为一个服务，用一个字符串和一对严格规范的消息定义：一个用于请求，一个用于回应。这类似于web服务器，web服务器是由URIs定义的，同时带有完整定义类型的请求和回复文档。需要注意的是，不像话题，只有一个节点可以以任意独有的名字广播一个服务：只有一个服务可以称之为“分类象征”，比如说，任意一个给出的URI地址只能有一个web服务器。

在上面概念的基础上，需要有一个控制器可以使所有节点有条不紊的执行，这就是一个ROS的控制器（ROS Master）。

ROS Master 通过RPC（Remote Procedure Call Protocol，远程过程调用）提供了登记列表和对其他计算图表的查找。没有控制器，节点将无法找到其他节点，交换消息或调用服务。

比如控制节点订阅和发布消息的模型如下：

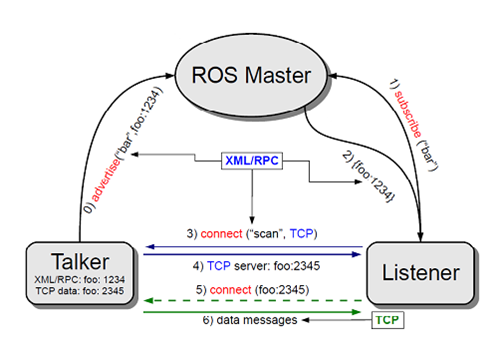


图 3.10

ROS的控制器给ROS的节点存储了主题和服务的注册信息。节点与控制器通信从而报告它们的注册信息。当这些节点与控制器通信的时候，它们可以接收关于其他以注册及节点的信息并且建立与其它以注册节点之间的联系。当这些注册信息改变时控制器也会回馈这些节点，同时允许节点动态创建与新节点之间的连接。

节点与节点之间的连接是直接的，控制器仅仅提供了查询信息，就像一个DNS服务器。节点订阅一个主题将会要求建立一个与出版该主题的节点的连接，并且将会在同意连接协议的基础上建立该连接。

另：ROS控制器控制服务：

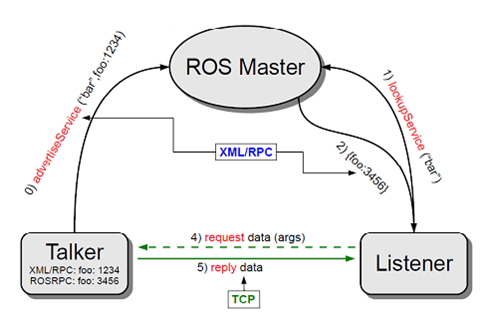
****

图 3.11

2.3.2 文件系统级

ROS文件系统级指的是在硬盘上面查看的ROS源代码的组织形式。

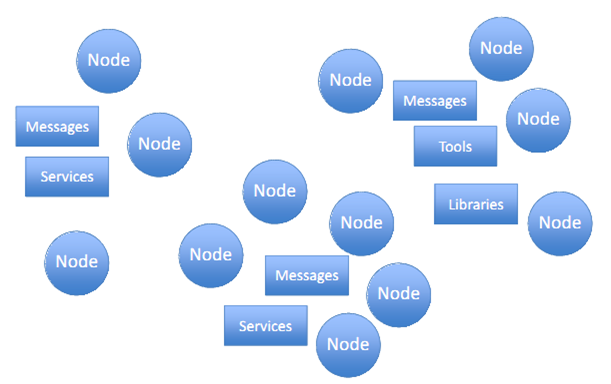


图 3.12

ROS中有无数的节点、消息、服务、工具和库文件，需要有效的结构去管理这些代码。在ROS的文件系统级，有以下几个重要概念：包（package）、堆（stack）等。

（1）包：



图 3.13

ROS的软件以包的方式组织起来。包包含节点、ROS依赖库、数据套、配置文件、第三方软件、或者任何其他逻辑构成。包的目标是提供一种易于使用的结构以便于软件的重复使用。总得来说，ROS的包短小精干。

（2）堆

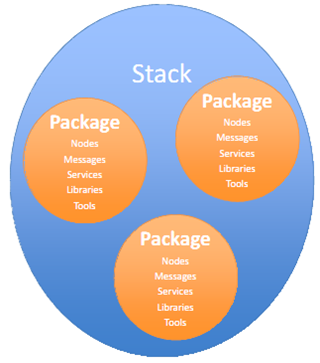


图 3.14

堆是包的集合，它提供一个完整的功能，像“navigation stack”。Stack与版本号关联，同时也是如何发行ROS软件方式的关键。

ROS是一种分布式处理框架。这使可执行文件能被单独设计，并且在运行时松散耦合。这些过程可以封装到包（Packages）和堆（Stacks）中，以便于共享和分发。下图是在包和堆在文件中的具体结构：

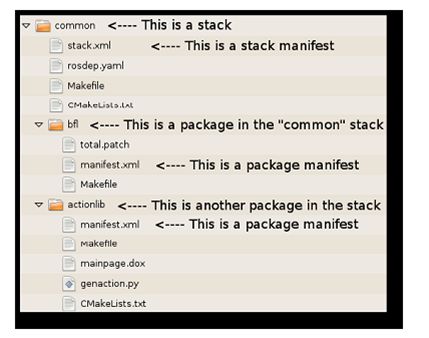


图 3.15

Manifests (manifest.xml)：提供关于Package元数据，包括它的许可信息和Package之间依赖关系，以及语言特性信息像编译旗帜（编译优化参数）。

Stack manifests (stack.xml)：提供关于Stack元数据，包括它的许可信息和Stack之间依赖关系。

2.3.3 社区级

ROS的社区级概念是ROS网络上进行代码发布的一种表现形式。结构如下图所示：

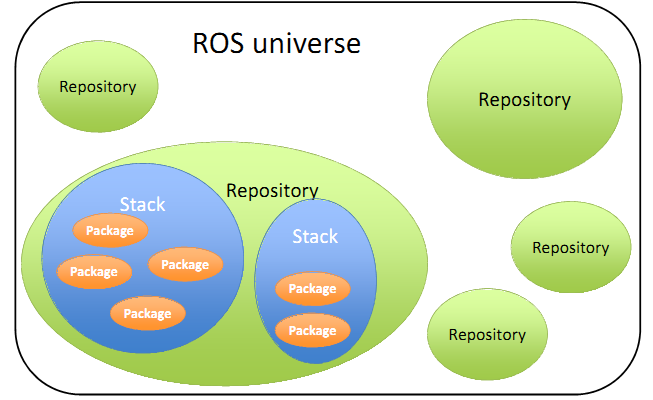


图 3.16

代码库的联合系统。使得协作亦能被分发。这种从文件系统级别到社区一级的设计让独立地发展和实施工作成为可能。正是因为这种分布式的结构，使得ROS迅速发展，软件仓库中包的数量指数级增加。

# 目标跟踪技术

## 3.1 粒子滤波理论

粒子滤波的想法来源于贝叶斯估计和蒙特卡罗方法，通过非参数化的蒙特卡罗模拟方法来实现递推贝叶斯滤波。适用于任何能用状态空间模型描述的非线性系统，精度可以逼近最优估计。粒子滤波器具有简单、易于实现等特点，它为分析非线性动态系统提供了一种有效的解决方法，从而引起目标跟踪、信号处理以及自动控制等领域的广泛关注。贝叶斯估计的基本原理是从状态变量的后验概率密度中获取状态的最优估计；而蒙特卡罗方法是根据对概率模型或随机过程的抽样试验其参数的特征值，继而获取此状态的无偏估计。

本节主要介绍了粒子滤波理论，动态空间模型，贝叶斯估计理论及蒙特卡罗方法，为粒子滤波算法在目标跟踪中的实现提供了理论依据。

3.1.1 动态空间模型

首先我们的机器人是一个动态的空间模型，那么如何来数学方法来描述这个模型就是一个很重要的问题。我们通常用状态空间模型的方法来描述动态空间，此状态的空间模型为：系统的状态是随着时间的变化来描述的状态转移模型和与系统状态相关的观测模型。在目标跟踪问题中，我们通过下面俩个方程来描述一个连续的非线性且非高斯的动态系统：

 （3-1）

其中分别为状态转移方程与观测方程，为系统状态，为观测值，为过程噪声，为观测噪声。为了描述方便，用与分别表示0到时刻所有的状态与观测值。在处理目标跟踪问题时，通常假设目标的状态转移过程服从一阶马尔可夫模型，即当前时刻的状态只与上一时刻的状态有关。另外一个假设为观测值相互独立，即观测值只与时刻的状态有关。

3.1.2 贝叶斯估计理论

贝叶斯统计是根据先验分布的方法来进行统计算，而贝叶斯估计是根据贝叶斯统计方法进行的预测。贝叶斯理论的实质是根据系统的先验概率，使用实时的观测值对其进行修正，继而得到系统的后验概率密度。因此贝叶斯滤波为非线性系统的状态估计问题提供了一种基于概率分布形式的解决方案。贝叶斯滤波将状态估计视为一个概率推理过程，即将目标状态的估计问题转换为利用贝叶斯公式求解后验概率密度或滤波概率密度，进而获得目标状态的最优估计。

贝叶斯滤波包含预测和更新两个阶段，预测过程利用系统模型预测状态的先验概率密度，更新过程则利用最新的测量值对先验概率密度进行修正，得到后验概率密度。

假设已知时刻的概率密度函数为，贝叶斯的具体过程如下：

1. 预测过程：其先验概率密度函数为，然后得到后验概率密度函数：

 （3-2）

（2）更新过程：根据系统观测值来修正先验概率密度函数，继而得到后验概率密度函数，递归公式：

 （3-3）

其为归一化常数：

 （3-4）

贝叶斯滤波需要进行积分运算，除了一些特殊的系统模型（如线性高斯系统，有限状态的离散系统）之外，对于一般的非线性、非高斯系统，贝叶斯滤波很难得到后验概率的封闭解析式。因此，现有的非线性滤波器多采用近似的计算方法解决积分问题，以此来获取估计的次优解。解决这个问题的一种行之有效的方法便是蒙特卡洛模拟方法。

3.1.3 蒙特卡罗方法

蒙特卡罗方法（Monte Carlo）,或称计算机随机模拟方法，是一种基于随机数的计算方法。这一方法源于美国在第一次世界大战中研制原子弹的“曼哈顿计划”。该计划的主持人之一、数学家冯·诺伊曼用驰名世界的赌城--摩纳哥的Monte Carlo来命名这种方法,为它蒙上了一层神秘色彩。传统的经验方法由于不能逼近真实的物理过程,很难得到满意的结果,而蒙特卡罗方法由于能够真实地模拟实际物理过程,能够得到令人满意的结果,这也正是该方法得到广泛应用的原因。

蒙特卡罗方法的基本原理如下：

当所要解决的问题是某种事件出现的概率，它们可以通过某种“试验”的方法来得到其出现的频率，并用它们作为问题的解。这就是蒙特卡罗的基本思想。它是以一个概率模型为基础,按照这个模型所描绘的过程,通过模拟实验的结果,作为问题的近似解。可以把蒙特卡罗解题归为三个主要步骤:构造或描述概率过程，实现从己知概率分布抽样;建立各种估计量。

蒙特特卡罗解决问题的三个步骤：

1. 构造或描述概率过程

对于本身就具有随机性质的问题,如粒子更新问题,主要是正确描述和模拟这个概率过程,对于本来不是随机性质的确定性问题,比如计算定积分,就必须事先构造一个人为的概率过程,它的某些参量正好是所要求解问题的解。即要将不具有随机性质的问题转化为随机性质的问题。

（2）实现从已知概率分布抽样

构造了概率模型以后,由于各种概率模型都可以看作是由各种各样的概率分布构成的,因此产生己知概率分布的随机变量(或随机向量),就成为实现蒙特卡罗方法模拟实验的基本手段,这也是蒙特卡罗方法被称为随机抽样的原因。

（3）建立各种估计量

一般来说,构造了概率模型并能从中抽样后,即实现模拟实验后,就要确定一个随机变量,作为所要求解问题的解,我们称它为无偏估计。建立各种估计量,相当于对模拟实验的结果进行考察和登记,从中得到问题的解。蒙特卡罗方法与一般计算方法有很大区别,一般计算方法对于解决多维或因素复杂的问题非常困难,而蒙特卡罗方法对于解决这方面的问题却比较简单。其特点如下:

1)直接追踪粒子,物理思路清晰,易于理解。

2)采用随机抽样的方法,较真切的模拟粒子运动的过程。

3)不受系统多维、多因素等复杂性的限制,是解决复杂系统问题的好方法。

4)程序结构清晰简单,应用灵活性强。

**3.2 粒子滤波算法**

粒子滤波又称序贯蒙特卡罗方法，是一种基于蒙特卡罗方法和递推贝叶斯估计的统计滤波方法，它依据大数定理，采用蒙特卡罗方法来求解贝叶斯估计中的积分运算。

粒子滤波算法首先依据系统状态向量的经验条件分布在状态空间产生一种随机样本的集合，称这些样本为粒子，然后根据观测量不断调整粒子的权重和位置，通过调整后粒子的信息修正先前的经验条件分布。当样本量足够大时，这种蒙特卡罗描述就近似于状态变量真实的后验概率密度。

粒子滤波适用于能用任何状态空间模型表示的非线性非高斯的随机系统，它完全突破了传统的Kalman滤波框架，对系统的过程噪声和测量噪声没有任何限制，可用于任何非线性系统，精度可以逼近最优估计，是一种很有效的非线性滤波技术，广泛用于数字通信，金融领域数据分析，统计学，图像处理，计算机视觉，自适应估计，语音信号处理，机器学习等领域。

3.2.1 粒子滤波基本原理

基于蒙特卡罗方法，如果状态变量为，已知其后验概率密度函数为，对于任意函数，可用如下公式来表示数学期望值：

 （3-5）

这时期望近似值可以表示为：

 （3-6）

式中={i=0,1....N}表示一系列离散样本,它是从后验概率密度函数中产生的,如果有足够多的样本,那么期望近似值绝对收敛于数学期望。

转换到粒子滤波器中,后验概率密度函数是用若干个粒子来近似的,并且釆用越多的粒子,近似也就越精确。由于后验概率分布很难直接求得,因此这成为解决问题的关键。

3.2.2 序列重要性采样和计算权值

序列重要性采样是创建粒子滤波器的基础,通常每个粒子的权重都是重要性采样来决定的,序列重要性采样可以通过不同的变换得出各种粒子滤波的研究算法。其基本思想是样本的后验概率密度用加权和表示,利用求和形式得出状态估计值。

假设在时刻，为此刻的系统状态变量，状态变量的后验概率密度函数用表示，其随机样本表示为表示样本粒子集，相对于每个粒子的权重用表示，归一化后为，则系统后验概率密度函数的近似值可表示为：

 （3-7）

其中为狄拉克函数，根据序列重要性采样的基本原理选择权重，如果后验概率密度函数是由粒子集得到的，则其权值表示为：

 （3-8）

若重要密度能分解为

 （3-9）

则通过由得到的粒子和由得到的粒子集可以得到新的粒子集。

由于后验概率密度函数可表示为





（3-10）

将式（3-8）、式（3-9）代入式（3-10），即可得到重要性权值更新公式为



（3-11）

如果，则重要密度函数仅依赖于和，在计算时仅需存储粒子，而不必关心粒子集和过去测量值。修正后的权值为：

 （3-12）

标准粒子滤波算法选择最易于实现的先验概率密度作为重要密度函数，即

 （3-13）

将（3-13）代入式（3-12），可将重要性权值化简为

 （3-14）

将归一化，即

 （3-15）

而后验概率密度可表示为

 （3-16）

式中，权值如（3-15）所示。可见当时，由大数定理即可保证上式可逼近真实后验概率。

标准粒子滤波算法如下：

1. 初始化。由先验概率产生粒子群，所有粒子权值为。
2. 更新。在时刻，更新粒子权值

 （3-17）

并且归一化

 （3-18）

则可得时刻未知参数的最小均方估计为

 （3-19）

（3）重采样。得到新的粒子集合。

（4）预测。利用状态方程预测未知参数。

（5）时刻，转到第（2）步。

但是，序列重要性采样会导致粒子退化现象的产生，也就是说，权值大的粒子个数很少，这就意味着随着时间推移，大多数的粒子对整个系统状态估计的作用会降到为零，继而产生粒子退化现象。

3.2.3 粒子退化和重采样方法

当用重要性函数替代后验概率分布作为采样函数时，理想情况是重要性函数非常接近后验概率分布，也就是希望重要性函数的方差基本为零。

但是，由于标准粒子滤波算法选择先验概率密度作为重要密度函数，若在对量测精度要求不高的场合，这种选取的方法能够获得较好的效果。不过，由于没有考虑当前的测量值，从重要性概率密度中取样得到的样本与从真实后验概率密度采样得到的样本有很大偏差。

因此，重要性权重的方差随着时间而随机递增，使得粒子的权重击中到少数粒子上，甚至在进过几步的递归之后，可能只有一个粒子有非零权值，其他粒子的权值很小，可以忽略不计，从而使大量的计算工作都被浪费在用下更新那些对的估计不起作用的粒子上，结果粒子集无法表达实际的后验概率分布，这就是粒子滤波算法的退化问题。

为了避免退化现象发生，我们这里引入了重采样技术，还有一种办法就是选取好的重要密度函数，下面我们介绍重采样技术。

设粒子和其权重组成二元数组，重采样的目的就是去掉权重较小的粒子，将较大权重对应的二元组分成若干个的二元组，可以通过从二元组集合中按照权值概率来采样得到新的二元组。

下面来描述一下重采样过程：首先计算粒子的概率累加和，并假设，随机采样第个服从[0,1]均匀分布的数，若，则可以取得第次随机采样的结果为，这样就达到了复制大权值粒子的效果。

重采样的基本思想是通过对后验概率密度重采样次，产生新的粒子集，使得。由于重采样是独立同分布的，权值被重新设置成=。

接下来就是如何确定退化程度呢，我们引入有效抽样尺度，定义为：

 （3-20）

式中，代表每个粒子占的权重，为的方差。一般无法计算的数值，但可以通过下式近似估计：

 （3-21）

由（3-20）可知，，越小，意味着退化现象越严重。增加粒子数目可以解决退化问题，但又使得计算量上升，影响算法的实时性。一般，粒子的数目由状态方程的维数、后验概率密度函数和重要密度函数的相似度及迭代次数决定。因此通常要考虑与其他策略结合，一般是在SIS之后进行重要性样本的重采样，判断是否重采样的依据是粒子的退化程度。首先设定一个有效样本数作为阈值，当<时，则进行重采样，这样就无需在每个时刻都进行重采样，从而能够自适应的根据样本情况决定是否进行重采样，可以在一定程度上降低算法复杂度。

重采样的副作用是样本枯竭，即有较大权值的粒子被多次选择，采样结果中包含了许多重复点，从而损失了粒子的多样性。这个问题还有待深入研究，也就是粒子滤波研究的重点。

3.2.4 粒子滤波算法的实现过程

根据序列重要性采样和重采样技术，我们得到了粒子滤波算法的简单步骤，其基本流程可以分为以下五个阶段：

1. 初始化。=0，采样~，即根据分布采样得到，。
2. 重要性采样和权值计算。设定，采样~。

计算重要性权值如下：





归一化重要性权值：

为后验概率估计提供样本值。

1. 重采样。根据粒子值的大小进行重采样，将权值小的粒子抛弃，保留权值大的粒子并替代被抛弃掉的粒子，得到新的样本集，再次将权值归一化。
2. 输出。得到新的粒子及其权值。

状态估计：

方差估计：

（5）判断是否结束，若是则退出本算法，否则返回步骤（2）。

由此可见粒子滤波算法结合了蒙特卡罗方法和贝叶斯理论的优点，用若干粒子作为系统随机抽取的样本，对状态的后验概率密度函数进行估计。假如有很多的粒子，则得到的状态后验分布的近似值就会很好。

**3.3 建立运动目标模型**

运动目标跟踪在军事制导，视觉导航，机器人，智能交通，公共安全等领域有着广泛的应用。例如，在车辆违章抓拍系统中，车辆的跟踪就是必不可少的。在入侵检测中，人、动物、车辆等大型运动目标的检测与跟踪也是整个系统运行的关键所在。所以，在计算机视觉领域目标跟踪是一个很重要的分支。

运动目标跟踪主流算法主要基于以下两种思路：

1. 不依赖于先验知识，直接从图像序列中检测到运动目标，并进行目标识别，最终跟踪感兴趣的目标。
2. 依赖于目标的先验知识，首先为运动目标建模，然后在图像序列中找到匹配的运动目标。

第（1）种方法需要先检测运动目标，这也是目前计算机视觉中研究的一个重点，在这里我就不叙述了。我这里用到的时第（2）种方法，即先为目标建模，然后匹配。

目标跟踪过程中必不可少的就是提取目标特征，基于特征匹配的跟踪方法不考虑运动目标的整体特征，只通过目标图像的一些显著特征来进行跟踪。假定运动目标可以由惟一的特征集合表达，搜索到该相应的特征集合就认为跟踪上了运动目标。除了用单一的特征来实现跟踪外，还可以采用多个特征信息融合在一起作为跟踪特征。基于特征的跟踪主要包括特征提取和特征匹配两个方面。

（1）特征提取

特征提取是指从景物的原始图像中提取图像的描绘特征，理想的图像特征应具备的特点是：

 a)特征应具有直观意义，符合人们的视觉特性;

b)特征应具备较好的分类能力，能够区分不同的图像内容;

c)特征计算应该相对简单，以便于快速识别;

d)特征应具备图像平移、旋转、尺度变化等不变性。

1. 特征匹配

特征提取的目的是进行帧间目标特征的匹配，并以最优匹配来跟踪目标。常见的基于特征匹配的跟踪算法有基于二值化目标图像匹配的跟踪、基于边缘特征匹配或角点特征匹配的跟踪、基于目标灰度特征匹配的跟踪、基于目标颜色特征匹配的跟踪等。基于特征的跟踪算法的优点在于对运动目标的尺度、形变和亮度等变化不敏感，即使目标的某一部分被遮挡，只要还有一部分特征可以被看到，就可以完成跟踪任务；另外，这种方法与 Kalman滤波器联合使用，也具有很好的跟踪效果。但是其对于图像模糊、噪声等比较敏感，图像特征的提取效果也依赖于各种提取算子及其参数的设置，此外，连续帧间的特征对应关系也较难确定，尤其是当每一帧图像的特征数目不一致、存在漏检、特征增加或减少等情况。

3.3.1 建立图像的颜色模型

根据前文所述我们应该先创建一个空间模型，这是因为目标的位置，速度等特征不能从图像中直接观测出来。基于自然界中各种的色彩，计算机视觉中常用的颜色空间模型有RGB空间，HSV空间。

1. RGB空间模型

最典型、最常用的面向硬件设备的彩色模型是三基色模型，即RGB模型。电视、摄像机和彩色扫描仪都是根据RGB模型工作的。每个颜色通道为0~255的像素值，可以通过三个通道之间的叠加变化得到种想要的颜色。

RGB颜色模型建立在笛卡尔坐标系统里，其中三个坐标轴分别代表R、G、B，如图3.1所示，RGB模型是一个立方体，原点对应黑色，离原点最远的顶点对应白色。RGB是加色，是基于光的叠加的，红光加绿光加蓝光等于白光。



图3.1 RGB颜色模型

（2）HSV空间模型

HSV彩色模型是从CIE三维颜色空间演变而来。在HSV彩色模型中，每一种颜色都是由色调（Hue，简H），饱和度（Saturation，简S）和亮度值（Value，简V）所表示的。

HSV彩色模型是一个倒立的六菱锥，如图3.2所示，不含黑色的纯净颜色都处于六菱锥顶面的一个色平面上。在HSV六菱锥色彩模型中，色相H处于平行于六菱锥顶面的色平面上，它们围绕中心轴V旋转和变化，红、黄、绿、青、蓝、品红六个标准色分别相隔60度。色彩明度沿六菱锥中心轴V从上至下变化，中心轴顶端呈白色V = 1，底端呈黑色V = 0，它们表示无彩色系的灰度颜色。色彩饱和度S沿水平方向变化，越接近六菱锥中心轴的色彩，饱和度越低。六边形正中心的色彩饱和度为零S = 0，与最高明度的V = 1相重合，最高饱和度的颜色则处于六边形外框的边缘线上S = 1。由于HSV颜色模型所代表的颜色域是CIE色度图的一个子集，它的最大饱和度的颜色的纯度值并不是100％。

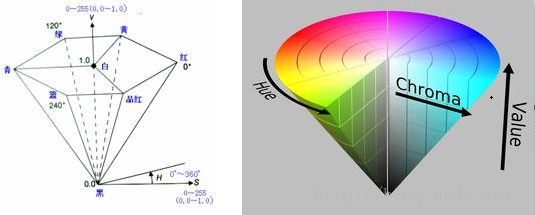


图3.2 HSV颜色模型

3.3.2 提取目标特征

目标跟踪中常用的运动目标的特征主要包括颜色、纹理、边缘、块特征、光流特征、周长、面积、质心、角点等。提取对尺度伸缩、形变和亮度变化不敏感的有效特征至今仍是图像处理研究领域中一个比较活跃的方面。

（1）颜色特征

颜色特征是一种全局特征,描述了图像或图像区域所对应的景物的表面性质。一般颜色特征是基于像素点的特征，此时所有属于图像或图像区域的像素都有各自的贡献。由于颜色对图像或图像区域的方向、大小等变化不敏感，所以颜色特征不能很好地捕捉图像中对象的局部特征。另外，仅使用颜色特征查询时，如果[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "http://blog.csdn.net/yeyang911/article/details/_blank)很大，常会将许多不需要的图像也检索出来。

1. 边缘特征

各种基于形状特征的检索方法都可以比较有效地利用图像中感兴趣的目标来进行检索，但它们也有一些共同的问题，包括：

①目前基于形状的检索方法还缺乏比较完善的数学模型；

②如果目标有变形时检索结果往往不太可靠；

③许多形状特征仅描述了目标局部的性质，要全面描述目标常对计算时间和存储量有较高的要求；

④许多形状特征所反映的目标形状信息与人的直观感觉不完全一致，或者说，特征空间的相似性与人视觉系统感受到的相似性有差别。另外，从2D图像中表现的3D物体实际上只是物体在空间某一平面的投影，从2D图像中反映出来的形状常不是3D物体真实的形状，由于视点的变化，可能会产生各种失真。

1. 纹理特征

纹理特征也是一种全局特征，它也描述了图像或图像区域所对应景物的表面性质。但由于纹理只是一种物体表面的特性，并不能完全反映出物体的本质属性，所以仅仅利用纹理特征是无法获得高层次图像内容的。与颜色特征不同，纹理特征不是基于像素点的特征，它需要在包含多个像素点的区域中进行统计计算。在模式匹配中，这种区域性的特征具有较大的优越性，不会由于局部的偏差而无法匹配成功。作为一种统计特征，纹理特征常具有旋转不变性，并且对于噪声有较强的抵抗能力。但是，纹理特征也有其缺点，一个很明显的缺点是当图像的分辨率变化的时候，所计算出来的纹理可能会有较大偏差。另外，由于有可能受到光照、反射情况的影响，从2D图像中反映出来的纹理不一定是3D物体表面真实的纹理。

例如，水中的倒影，光滑的金属面互相反射造成的影响等都会导致纹理的变化。由于这些不是物体本身的特性，因而将纹理信息应用于检索时，有时这些虚假的纹理会对检索造成“误导”。在检索具有粗细、疏密等方面较大差别的纹理图像时，利用纹理特征是一种有效的方法。但当纹理之间的粗细、疏密等易于分辨的信息之间相差不大的时候，通常的纹理特征很难准确地反映出人的视觉感觉不同的纹理之间的差别。

3.3.3 颜色直方图

颜色直方图是最常用的表达颜色特征的方法，其优点是不受图像旋转和平移变化的影响，进一步借助归一化还可不受图像尺度变化的影响，目前在目标跟踪中已经成为最常用的方法。

颜色直方图用以反映图像颜色的组成分布，即各种颜色出现的概率。首先利用颜色空间三个分量的剥离得到颜色直方图，之后通过观察实验数据发现将图像进行旋转变换、缩放变换、模糊变换后图像的颜色直方图改变不大，即图像直方图对图像的物理变换是不敏感的。因此常提取颜色特征并用颜色直方图应用于衡量和比较两幅图像的全局差。另外，如果图像可以分为多个区域，并且前景与背景颜色分布具有明显差异，则颜色直方图呈现双峰形。

我们这里计算图像的颜色直方图需要将RGB颜色空间分成很多个小区间，用Bin来表示。一般我们将RGB空间分成个区间，根据每种颜色出现的频数得到这幅图像的颜色直方图。

如果RGB图像的像素总共有N个，其中一种颜色的像素个数为，则对图像的颜色特征归一化后得出：

 （3-22）

则颜色直方图就可以表示为：

 （3-23）

图像之间的相似度是由颜色直方图来决定的，当直方图出现零点时就不能表现出图像间的区别，这时我们通过直方图的累加来使图像之间的差别更明显。用表示图像的一般直方图，令，则图像的累加直方图为：

 （3-24）

3.3.4 基于颜色特征的相似度计算

图像相似度计算主要用于对于两幅图像之间内容的相似程度进行打分，根据分数的高低来判断图像内容的相近程度。可以用于计算机视觉中的检测跟踪中目标位置的获取，根据已有模板在图像中找到一个与之最接近的区域,然后一直跟着。已有的一些算法比如BlobTracking，Meanshift，Camshift，粒子滤波等等也都是需要这方面的理论去支撑。

在运动目标跟踪过程中，一般要根据当前的系统状态得到一个预测值，然后用当前值与预测值进行比较修正，最后用修正后的值与已经选取的目标模板进行相似度计算。因此需要设计一个相似度函数，将已知图像特征与当前图像特征区分开来。比如有图像A和图像B，分别计算两幅图像的直方图，HistA，HistB，然后计算两个直方图的归一化相关系数（巴氏距离，欧氏距离）等等。

本文选用巴氏系数来表示俩幅图像目标区域的相似度，这种思想是基于简单的数学上的向量之间的差异来进行图像相似程度的度量，这种方法是目前用的比较多的一种方法，因为直方图能够很好的归一化，比如通常的256个Bin条的。那么两幅分辨率不同的图像可以直接通过计算直方图来计算相似度很方便，而且计算量比较小。

系数：

 （3-25）

巴氏系数是对两个统计样本的重叠量的近似计算，可用来对两组样本的相关性进行测量。式中，表示所选区域的的已知目标模板，表示预测区域的特征模板，则所选目标区域的颜色直方图的巴氏系数可由上式表示，可以看出的值与颜色直方图的相似性成正比关系，越大相似度越高。

目标已知的当前状态与预测状态之间的相似度函数可表示为：

 （3-26）

式（3-26）中巴氏距离越小，说明当前状态与预测状态之间的相似度越大，在粒子滤波中更新的粒子的权值也就越大，这样粒子稳定性越强。

**3.4 融合Kinect的深度信息**

本文选用的是基于颜色信息的RGB颜色模型作为目标跟踪的特征模板，这里就会出现一个问题，就是与目标颜色相近的物体会干扰系统的模型建立。尤其是在干扰物与目标在图像上位置相近时。为了改善这一问题，我们可以将Kinect获取的深度信息融合到采样特征与特征模板的相似度计算中，通过实验此方法具有一定的有效性。

在算法中使用数量为N的粒子集。其中表示第个粒子在图像上的坐标；表示当前区域与初始选取的目标区域大小的比例尺；代表粒子的权重，代表粒子所在图像上的深度值。

定义为距离相似度函数，当粒子深度信息与参考距离相同时其值为1，随着俩个距离之间的增大，其值逐渐向0靠近，下面叙述的算法通过该距离相似度函数对传统的权值进行扩展，从而达到融合深度信息的目的。

融合深度信息的算法流程：

1. 选取目标，计算目标区域的颜色直方图并初始化粒子集，其中的初始值设为选取的目标区域的中心；的值设为1，的初始值设定为中心的深度值，同时为参考距离赋值。
2. 粒子采样，根据高斯分布对当前粒子采样，得到新的和。
3. 计算权值，根据采样得到的新和计算当前粒子对应图像区域的颜色直方图并与（1）中获取的颜色直方图模板进行比较,釆用公式计算出相应的权值，再利用深度信息对权值进行修改并进行归一化处理。

由巴氏距离得到权值之后，根据粒子的深度信息与参考距离，对进行更新，则新的为：

 （3-27）

然后再对权值进行归一化处理，令。

1. 根据公式计算目标的最优估计位置，同时计算估计位置在图像上的颜色直方图和深度值，与参考直方图和参考距离进行比较，如果其与参考模板的差异在一定的阈值之内，则确认目标跟踪成功，并以当前的深度值更新参考距离。
2. 重采样，根据权值的大小舍弃一部分粒子。并依据权值较大的粒子衍生出新的粒子，然后回到第（2）步。

下面是各种算法的比较：

# 路径规划技术

**4.1 常用的路径规划技术**

路径规划技术机器人研究领域中的一个重要分支。所谓机器人的最优路径规划问题，就是依据某个或某些优化准则（如工作代价最小，行走路线最短，行走时间最短等），在其工作空间中找到一条从起始状态到目标状态的能避开障碍物的最优路径。

那么怎样找到这条最优路径呢？我们可以依据某种最优准则，在工作空间中寻找一条从起始状态到目标状态的避开障碍物的最优路径。这时我们需要解决的问题是：

1. 始于起始点止于目标点。
2. 壁障。
3. 尽可能优化的路径（包括时间最优或者空间最优）。

4.1.1 路径规划的分类

按照不同的分类方法，得到的分类也是不一样的。我们先看看下面这张图，大概了解一下路径规划的发展史。

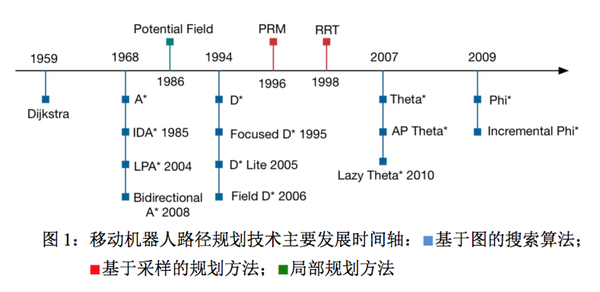


图4.1

根据上图我们可以有如下分类：

1. 静态结构化环境下的路径规划
2. 动态已知环境下的路径规划
3. 动态不确定环境下的路径规划

也可以有：

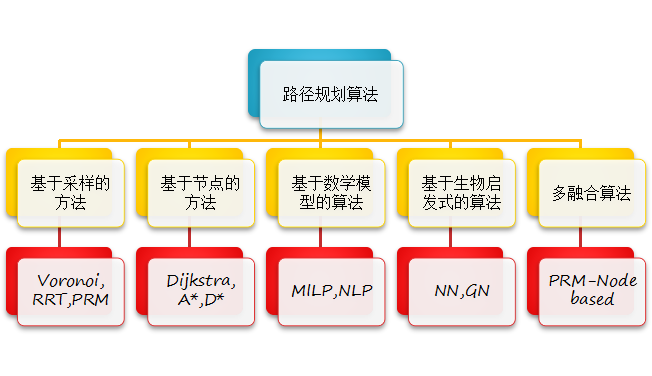


图4.2

还可以有：

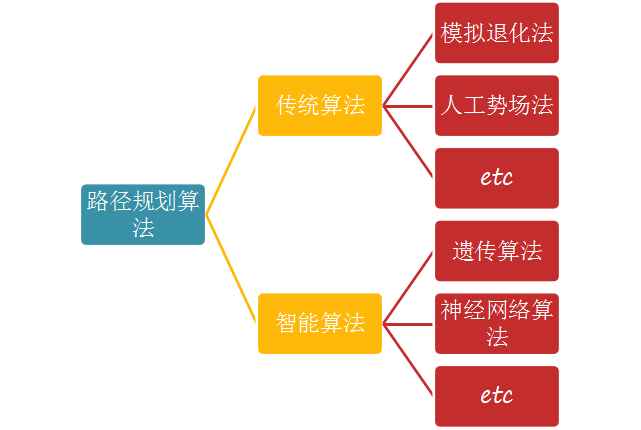


图4.3

4.1.2 快速扩展随机树算法

2000年以来，随机化方法越来越多的被用到路径规划算法中，由于其随机性的特点，这类方法可适用于不确定环境中的高纬度空间，不会产生局部环境迷失和计算量大等问题，进而被广泛应用。快速扩展随机树算法（Rapidly Exploring Random Trees,RRT）是随机化算法中典型的一个。由 LaValle 和 Kuffner 在 1999 年的 IEEE 国际会议上首次提出。它不同于传统的基于反应模式或基于仿生学的算法,首次将随机化方法引入到轨迹规划中,由于具有随机性的点,RRT 算法可以很好地解决在不确定环境下的完整性规划，非完整性规划以及运动动力学的问题。RRT算法的基本思想是通过树杈延伸的方式逐步的降低树结构与目标点之间的距离，直至树杈达到目标点或目标点的阈值范围内，最后用树的脉络来表示从起始点到目标点的路径。

RRT算法的流程如下图4.10所示。首先对环境建模，设起始点（树根）为，T为树结构，为状态空间。RRT路径规划实际上就是从中寻找出一条从到的路径，其过程为遍历T，找到T上距离最近的节点，然后在工作空间内随记的选择下一个叶节点，机器人沿着到达，则可以作为树结构上新的树杈点，以此作为下次迭代过程的，重复上述过程，逐步向靠近，直到T达到目标点或目标点的阈值范围内，随机树构建结束。

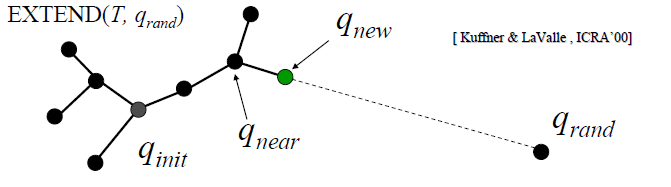


图4.10

由以上算法可以看出影响RRT性能的主要因素有：目标点的数量，随机扩展树的步长，随机点的分布规律以及已知树结构叶点的数量。

该方法的缺点是：新叶点的产生是随机的，树的延生具有随机性，导致了轨迹规划的随机性，为构建地图而进行的没有针对性和重复性的探索。

4.1.3 蚁群规划算法

蚁群算法(Ant Algorithm简称AA)是近年来刚刚诞生的随机优化方法，它是一种源于大自然的新的仿生类算法。由意大利学者Dorigo最早提出，蚂蚁算法主要是通过蚂蚁群体之间的信息传递而达到寻优的目的，最初又称蚁群优化方法(Ant Colony Optimization简称ACO)。由于模拟仿真中使用了人工蚂蚁的概念，因此亦称蚂蚁系统(Ant System，简称AS)。

蚁群算法是利用群集智能（swarm intelligence）解决组合优化问题的典型例子，作为一种新的仿生类进化算法，该算法模仿蚂蚁觅食时的行为，按照启发式思想，通过信息传媒—菲洛蒙(Pheromone)的诱导作用，逐步收敛到问题的全局最优解，迄今为止，蚂蚁算法己经被用于TSP问题，随后应用在二次分配问题(QAP)、工件排序问题、车辆调度等问题。

ACO优化机理：蚂蚁有能力在没有任何提示下找到从其巢穴到食物源的最短路径，并且能随环境的变化而变化，适应性的搜索新的路径，产生新的选择。其根本原因是蚂蚁在寻找食物源时，能在其走过的路上释放特殊的信息素(Pheromone)，随着时间的推移该物质会逐渐挥发，后来的蚂蚁选择该路径的概率与当时这条路径上该物质的强度成正比。当一定路径上通过的蚂蚁越来越多时，其留下的信息素轨迹也越来越多，后来蚂蚁选择该路径的概率也越高，从而更增加了该路径的信息素强度，而强度大的信息素会吸引更多的蚂蚁，从而形成一种正反馈机制。通过这种正反馈机制，蚂蚁最终可以发现最短路径。特别地，当蚂蚁巢穴与食物源之间出现障碍物时，蚂蚁不仅可以绕过障碍物，而且通过蚁群信息素轨迹在不同路径上的变化，经过一段时间的正反馈，最终收敛到最短路径上。

蚁群算法在优化过程中的俩个重要规则

1. 蚂蚁在众多路径中转移路线的选择规则:蚂蚁倾向于选择信息素浓度高的路径，信息素类似于一种分布式的长期记忆，它不是局部地存在于单个的蚂蚁上，而是全局地分布在整个问题空间中，这就形成了一种间接联络方式。
2. 全局化信息素更新规则:路径段上的信息素浓度，一部分会因自然蒸发而逐渐减少，这样没有蚂蚁走过的路径上的信息素浓度会由于自然蒸发而越来越低，从而变得越来越不受欢迎。每只蚂蚁按路径的长短成比例地在其经过的路径上留下一定数量的信息素。信息素更新的实质就是人工蚂蚁根据真实蚂蚁在访问过的边上留下的信息素和蒸发的信息素来模拟真实信息素数量的变化，从而使得越好的解得到越多的增强。这就形成了一种自催化强化学习(Autocatalytic Reinforcement Learning)的正反馈机制。

4.1.4 人工势场法

人工势场法是由Khatib提出的一种虚拟力法。原理是：将机器人在环境中的运动视为一种机器人在虚拟的人工受力场的运动。障碍物对机器人产生斥力，目标点对机器人产生引力，引力和斥力的合力作为机器人的加速力，来控制机器人的运动方向和计算机器人的位置。

如图4.11所示，机器人在一个二维环境下运动，图中指出了机器人，障碍和目标之间的相对位置。

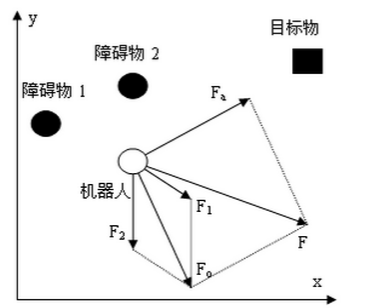


图4.11

人工势场包括引力场合斥力场，其中目标点对物体产生引力，引导物体朝向其运动（这一点有点类似于A\*算法中的启发函数h）。障碍物对物体产生斥力，避免物体与之发生碰撞。物体在路径上每一点所受的合力等于这一点所有斥力和引力的和。这里的关键是如何构建引力场和斥力场。下面我们分别讨论一下：

1. 引力场（attraction）：随机器人与目标点的距离增加而单调递增，且方向指向目标点；

常用的引力函数：

 （4-1）

这里的是尺度因子，表示物体当前状态与目标的距离。引力有了，那么引力就是引力场对距离的导数（类比物理里面W=FX）：

 （4-2）

引力场模型：

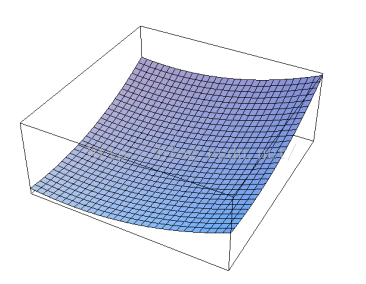


图4.12

1. 斥力场（repulsion）：在机器人处在障碍物位置时有一极大值，并随机器人与障碍物距离的增大而单调减小，方向指向远离障碍物方向。

常用的斥力函数：

 （4-3）

式中是斥力尺度因子，代表物体障碍物之间的距离。代表每个障碍物的影响半径。换言之，离开一定的距离，障碍物对物体没有斥力影响。

斥力就是斥力场的梯度：

（4-4）

斥力场模型：

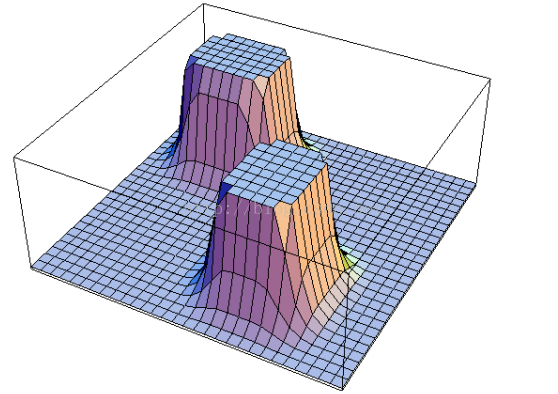


图4.13

1. 总场就是斥力场和引力场的叠加，也就是，总的力也是对应的分力的叠加，如图4.14 ：

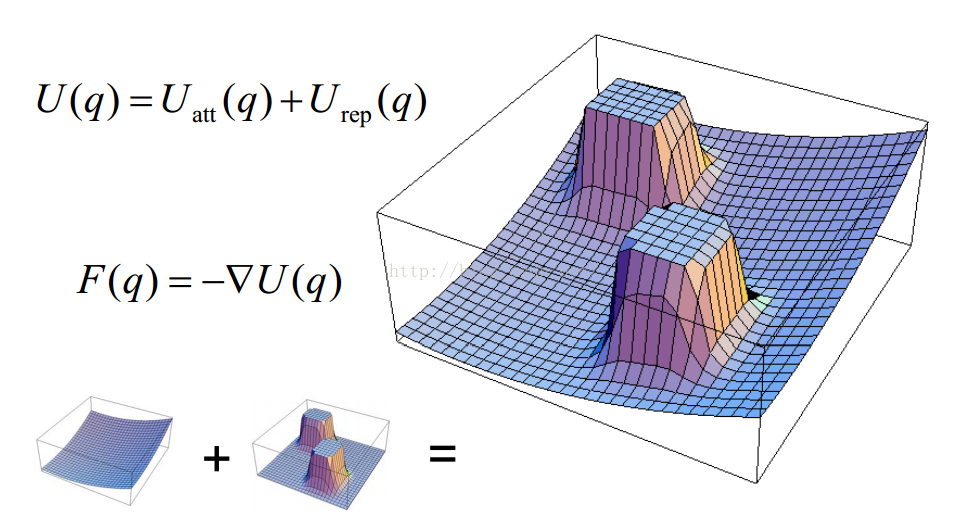


图4.14

当然，每种算法都有自己的优缺点，人工势场也存在：

1. 优点：

便于底层的实时控制，在实时避障和平滑的轨迹控制方面，得到了广泛应用

1. 缺点：

（a）当物体离目标点比较远时,引力将变的特别大，相对较小的斥力在甚至可以忽略的情况下，物体路径上可能会碰到障碍物。

（b）当目标点附近有障碍物时，斥力将非常大，引力相对较小，物体很难到达目标点。

（c）在某个点，引力和斥力刚好大小相等，方向想反，则物体容易陷入局部最优解或震荡。

**4.2 算法**

在计算机科学中，算法作为算法的扩展，因其高效性而被广泛应用于寻路及图的遍历，如星际争霸等游戏中就大量使用。在理解算法前，我们需要知道几个概念：

1. 搜索区域（The Search Area）

如图4.4中中的搜索区域被划分为了简单地二维数组，数组中每个元素对应一个小方格，当然我们也可以将区域等分成是五角星，矩形等。通常将一个单位的中心点称之为搜索区域节点（Node），而非方格（Squares）。

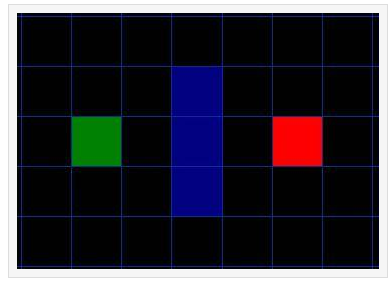


图4.4

1. 开放列表（Open List）

我们将路径规划过程中待检测的节点存放于Open List中，而已检测过的节点则存放于Close List中。

1. 父节点（Parent）

在路径规划中用于回溯的节点，开发时可考虑为双向链表结构中的父节点指针。

1. 路径排序（Path Sorting）

具体往哪个节点移动由以下公式确定：



G代表的是从初始位置A沿着已生成的路径到指定待检测节点的移动开销

H指的是待测节点到目标节点B的估计移动开销

1. 启发函数（Heuristics Function）

H为启发函数，也被认为是一种试探，由于在找到唯一路径前，我们不确定在前面会出现什么障碍物，因此用了一种计算H的算法，具体根据实际场景决定。在我们的简化模型中，H采用的是传统的曼哈顿距离（Manhattan Distance）,也就是横纵走向的距离之和。

算法流程：

我们假定图4.4中的绿色方块为机器人起始位置A。红色方块为目标位置B，蓝色为障碍物。现用算法寻找出一条自绿色A到红色B的最短路径，经简化，每个方格的边长我10，即水平垂直方向移动开销为10.因此斜对角移动开销约等于14。

（1）从节点A开始，搜索其临近节点，直到找到目标点。从节点A开始，把一系列待考虑的节点放入Open List里面，Open List存放着一系列需要检查的节点（方块），如图4.5，首先检查起点周围的8个节点.给每个节点赋值

F=G+H。

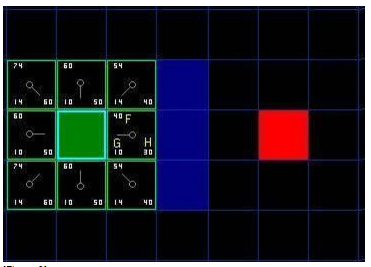


图4.5

1. 找到F值最小的节点作为新的起点，将它从Open Lsit中删除，加入到Close List里面，检查它的临近节点，忽略已经在Close List中的节点和不可行节点（障碍）。如果临近节点已经在Open List里面，则对比一下是否从现节点到临近节点的G值比原G值小，若是，把现节点作为父节点。否，不做改动。如图4.6

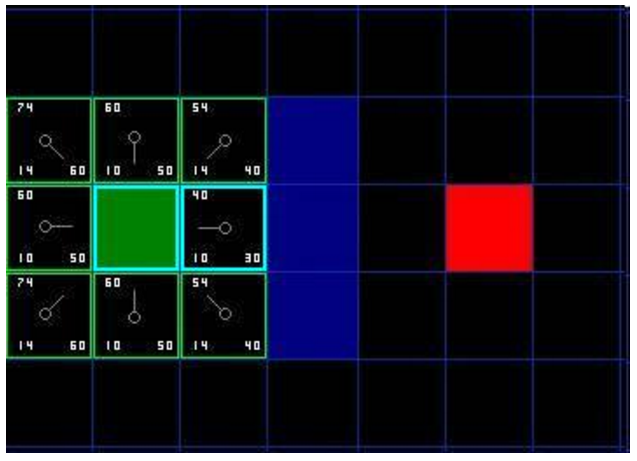
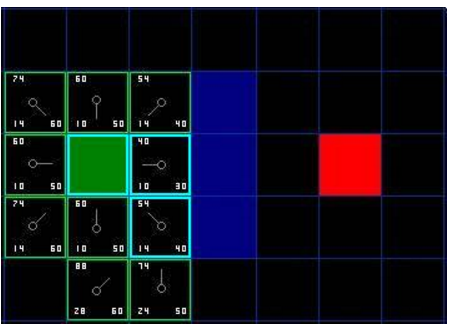


图4.6

（3）上步骤中新节点未造成任何改动，我们继续在Open List中寻找新的节点，如图4.7。重复（1）,（2）中的步骤，直到我们找到目标节点。



新节点

图4.7

1. 寻找到目标节点。如图4.8

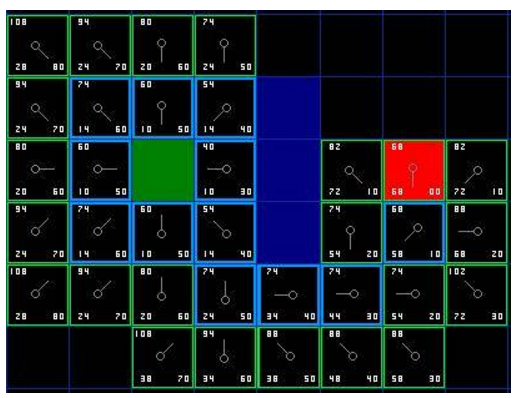


图4.8

（5）从目标节点回溯可以找到初始点，从而确定路径。如图4.9

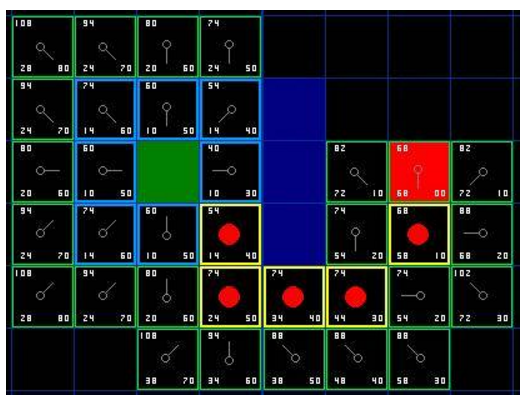


图4.9

但也发现，在整个计算过程中，A\*算法结合了启发式方法，利用估值函数F(H)来估计途中当前点与终点距离，并由此决定搜索方向，当这条路失败会重新尝试其他路径，但不理想的估值函数会导致整个算法运行很慢，而且，这种算法虽说在时间上最优，但也存在空间增长是指数级别的缺点，因此在往高维状态空间进行运算时，速度会受到影响，基于A\*算法迭代加深的IDA\*算法则有效解决了空间增长带来的问题。

**第5章 系统设计验证**

**5.1 系统软件设计**

**5.2 实验结果**

**结 论**

**参考文献**

**致 谢**