摘 要

在当今的21世纪，机器人和无人机发展成为社会的宠儿。这样的发展趋势越来越明显，如使用无人机相机进行自拍视频，非常有趣，还不需要花费很多，成本很低。

除了无人机以外，现在越来越多的土地机器人也开始“跟随领导”的功能，比如凯迪特雷克推出的电动高尔夫球车已经能够取代部分劳动力，因此可以看到追着你的机器人就在你身后。

本文实现了基于ROS系统的移动跟随机器人，通过手动选择“显眼”的目标，它会随着目标的移动而移动。这个系统最重要的是跟随执行算法，内核相关滤波算法近年来更好的跟踪结果，成功移植到了ROS，实现了目标跟随功能。

关键字 移动机器人；目标跟踪；核相关滤波；ROS机器人操作系统

**Abstract**

In the robot and the UAV in the world, a trend is growing steadily is to "follow" the user permission, the trend of the play is very interesting, such as the use of UAV GoPro camera self timer video collocation, not only fun, but also to spend lots of money to hire a helicopter and crew., low cost to many.

In addition to the UAV, now there are more and more land robots have also started to "follow the leader" function, such as CaddyTrek launched the electric golf caddy has been able to replace part of the labor force, so to be able to chase your shadow around the robot has been just around the corner.

This paper describes the design of a follower robot based on the ROS system, by choosing a "conspicuous" target, the robot will move with the movement of the target. This system is the most important thing is to follow the implementation of the algorithm, kernel correlation filtering algorithm in recent years better tracking results, successfully transplanted to ROS, to achieve the goal follow the function.

**Key words** Mobile robot Target tracking Kernel correlation filter

Robot Operating System

目 录

摘 要 I

Abstract II

[第1章 绪 论 1](#_Toc6686)

[1.1 引言 1](#_Toc1517)

[1.2 移动机器人发展史 1](#_Toc10268)

[1.3 ROS（Robot Operating System）简介 1](#_Toc23595)

[1.4 视觉目标跟踪技术 2](#_Toc19930)

[1.5 本文主要研究内容 2](#_Toc30247)

[第2章 机器人硬件系统 3](#_Toc20580)

[2.1 整体框架 3](#_Toc9903)

[2.2 M-robot机器人介绍 3](#_Toc6520)

[2.2.1 主控制器 4](#_Toc12811)

[2.2.2 电机驱动板介绍 4](#_Toc24509)

[2.2.3 电源板介绍 5](#_Toc20952)

[2.3 Kinect 6](#_Toc9632)

[第三章 ROS介绍 7](#_Toc27680)

[3.1 ROS概览 7](#_Toc16272)

[3.2 ROS特点 7](#_Toc8476)

[3.3 ROS总体框架 10](#_Toc24785)

[3.3.1 计算图级 10](#_Toc19673)

[3.3.2 文件系统级 12](#_Toc16059)

[2.3.3 社区级 13](#_Toc24788)

[第4章 目标跟踪技术 15](#_Toc1595)

[4.1 视觉目标跟踪概述 15](#_Toc23077)

[4.1.1 视觉跟踪介绍 15](#_Toc7288)

[4.1.2 视觉目标跟踪分类 16](#_Toc15356)

[4.2 核相关的视觉目标跟踪算法（KCF） 16](#_Toc2021)

[4.2.1 算法描述 16](#_Toc19198)

[4.2.2 线性回归 17](#_Toc23204)

[4.2.3 循环偏移 17](#_Toc30860)

[4.2.4 循环矩阵 18](#_Toc28985)

[4.2.5 整合公式 19](#_Toc9526)

[4.2.6 核技巧（Kernel trick）简要概述 20](#_Toc12193)

[4.2.7 快速核回归 20](#_Toc23536)

[4.2.8 快速检测目标 21](#_Toc30784)

[4.2.9 快速计算核函数相关性 22](#_Toc29635)

[4.2.10 多通道数据处理 22](#_Toc13141)

[4.3 KCF算法实现 22](#_Toc13313)

[4.3.1 视觉目标检测框架 22](#_Toc12995)

[4.3.2 软件设计 24](#_Toc14170)

[第五章 系统实现 28](#_Toc23925)

[5.1 实际场景运行 28](#_Toc27701)

[5.2 系统实验效果分析 29](#_Toc9904)

[结 论 31](#_Toc17727)

[参考文献 32](#_Toc14595)

[致 谢 33](#_Toc786)

# 

# 第1章 绪 论

## 1.1 引言

二十一世纪是什么样的世纪？是VR世纪的世纪吗？也许，但我也相信二十一世纪是机器人的世纪。目前，中国已进入经济转型的拐点，需要更多的产业来支持中国经济的发展。在许多行业已经开始机换机，升级生产步伐。中国多部门发布了机器人行业规划。因此未来发展空间是非常巨大。从服务机器人手中，还要满足未来市场需求的增长，一是包括养老金，老年人，残疾人等基本生活需要。二是救灾等国家安全需要，机器人，如娱乐，儿童教育，家居智能应用。

移动的机械是这个课题研究的重要成员。它需要有一个强大的“腿”，明亮的“眼睛”，聪明智能的“大脑”，实现这一切涉及到许多技术领域。

在本文中，介绍了视觉目标跟踪技术，智能移动机器人系统可以通过使用Kinect相机在复杂的环境中运行。

## 1.2 移动机器人发展史

到目前为止，在产业建造空间中，类人形的机械取得了很大的胜利。机器臂，机器手，焊接机械和机器人绘画组成了庞大的产业产出。然而，关于一切的胜利利用，会有一个一起的不足：没有跑动性。机械手安装在不变的地方，它的活动地方不大。相反，移动机器人那个移动到工厂的任何位置，最大化的做到了物尽其用。

六十年代末，美国和前苏联都在月球探测中应用到了移动机器人。移动机器人发挥了重要的作用。地面人员可以操作机器人在月球表面工作，它们可以做到对月球的挖掘活动和其它的任务。在月球上的机器人，采集土壤和岩石样品。70年代，早稻田大学发明了一个是人形的双足步行机械。为顺应社会的发展，有非凡能力的机器人和水下机器人展开起来，在各级政府的支持下，中国机器人的也取得了很大进展。

## 1.3 ROS（Robot Operating System）简介

ROS是一个分布式系统，它的很重要的特色是拥有节点。每个节点可以单独看作是一个独立的单元，它可以从外部接收消息，也可以在外面发布他们的消息。服务是实现节点功能的典型方式，正在接收外部信息，并对消息做出完全响应。以下是经典的例子：加法运算服务节点，定时查询消息，一旦收到消息（两个数字）立即返回，添加结果（和节点数）是一个单独的处理单元，它将大大简化程序设计和代码，代码层中的每个节点似乎是一个独立的可执行文件，一个main（）函数的入口。

另外，通过低层通信实现的通信机制，节点之间的通信类似于TCP，因此ROS上运行的相同计算平台上的节点不重要（不考虑运行能力），只要这些节点具有有效的通信通道，可以使相对计算平台透明化。这也是分布式系统的特征。一系列模拟，调试工具，如gazebo，rviz，rqt，更重要的是使用各种语言开发，如C ++，Python等。

## 1.4 视觉目标跟踪技术

在依据颜色信息的二维视频跟踪算法中。一般采取追踪目标或目标运动信息的办法。目标跟踪的本质实际上是模板匹配问题，此思路比较容易，咱们一定要追欧总物体存在的很好。那么在每一帧结束的时候，我们发现在整个形象中最相似，我们相信这是21世纪的目标，人们以概率来判断目标的匹配度的现代过滤方法，如常见的扩展卡尔曼滤波器，卡尔曼滤波器，粒子滤波器，相关滤波等技术。

## 1.5 本文主要研究内容

在本文中，基于以下机器人的主要技术难点，研究了目标跟踪的相关知识。

使用了叫做Kinect的深度摄像头。作为机器人采集信息的图像传感器。

为了实现视觉目标跟踪，我们讨论了KCF算法，将KCF算法集成到ROS系统中，实现后续系统的设计。

# 

# 第2章 机器人硬件系统

## 2.1 整体框架

为了实现目标跟踪和路径规划，我们使用Microsoft Corp Kinect作为移动机器人的“眼睛”，用于获取目标跟踪和位置信息。它是由M-robot机器人开发团队创造的机器人，是一个简单的移动机器人。

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，能够实现室内建立地图和导航。也能够作为ROS的学习平台。但无法达到消费品导航效果的稳定性，可以在开源算法的基础上继续优化。取得更好地效果。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）有各种接口：超声波。舵机接口。UART接口。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

主要资源有如下：

（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）板载的接口：有电源输入输出接口，USB，编码器，电机运动等的接口。

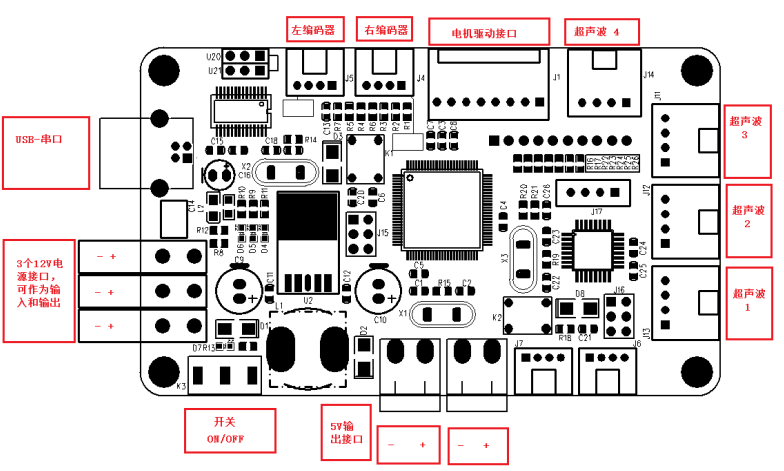


图2.2 主控制器

### 2.2.2 电机驱动板介绍

此款电机的运动控制板，能够当成直流电机驱动。它的理论电流100A。假如需要驱动大功率电机，必须加装散热器。

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。

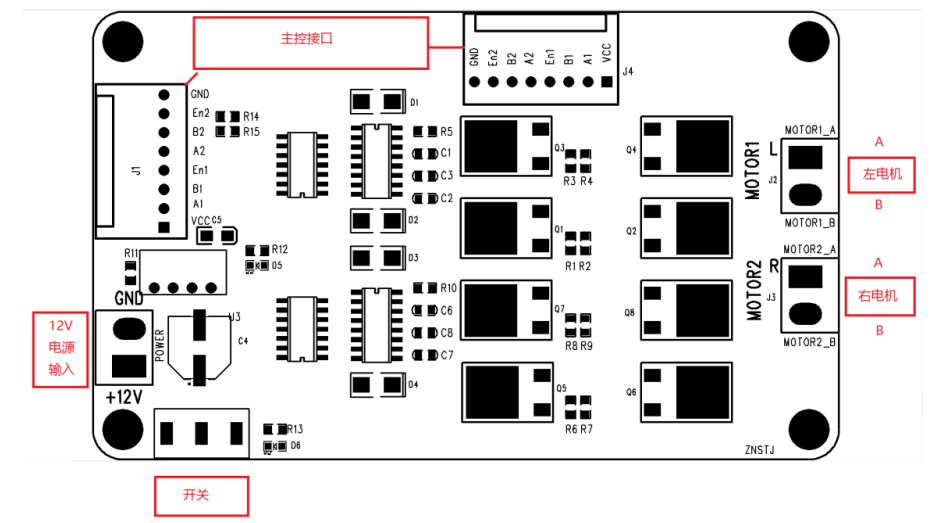


图2.3 电机驱动板

### 2.2.3 电源板介绍

电源板能够适应锂电池和蓄电池，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

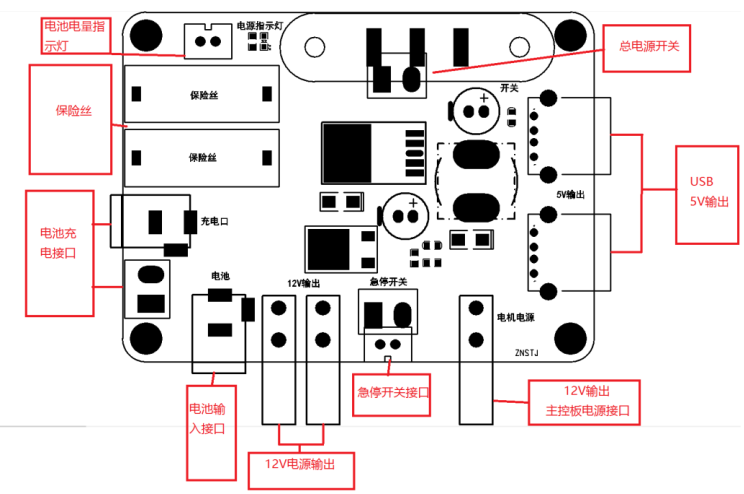


图2.4 电源板

## 2.3 Kinect

Kinect for Xbox 360，简称Kinect，由Microsoft开发，用于Xbox 360大型机外设。它能够不让游戏玩家或手持控制器在系统界面，或者是用语音命令或手势来操作Xbox360。2010年11月4日，在美国，这款149.Kinect在销售前60天销售了800万，目前已经有了ShenGuinness世界纪录，全球销量最快的消费电子产品。

2012年2月1日，微软正式发布了Windows系统Kinect版“Kinect for Windows”，即249美元。 2012年底，微软还将为用户教育版Kinect发行。

### 2.3.1 kinect硬件介绍

Kinect有三个镜头，中间镜头是RGB彩色相机，用于收集彩色图像。镜头的左右两侧有。红外发射器，红外CMOS摄像机的深度传感器，用来采集数据。最大支持1280 \* 960分辨率彩色摄像头，红外摄像机支持640 \* 480成像.Kiect搭配追求焦炭，电机将与移动底座一起旋转.Kinect还内置麦克风阵列，四路由迈克风同时对照去掉噪音，并可以用其获取语音辨认和声源确定位置。

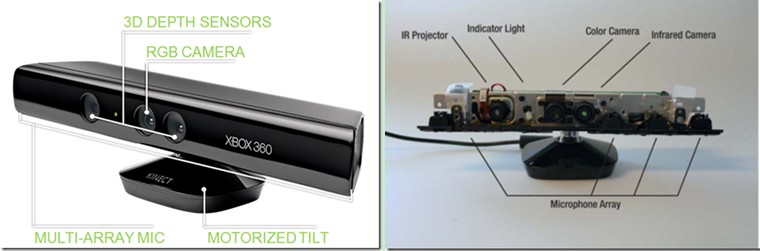


图2.5 kinect

# 

# 第3章 ROS介绍

## 3.1 ROS概览

伴着机器人领域的快速发展和复杂，可重复使用和模块化的要求是一个趋势。开放源码的机器人程序能够不错地满足需求。其中最有代表性当属近年来比较人们的ROS系统了。他受到了全球机器人爱好者的追捧，可以看做是机器人世界中的“安卓”操作系统。

## 3.2 ROS特点

ROS是开放源码的分开来的系统，能够把它当做机器人的操作系统。它有很好的底层硬件封装，也有上层成熟的应用包。

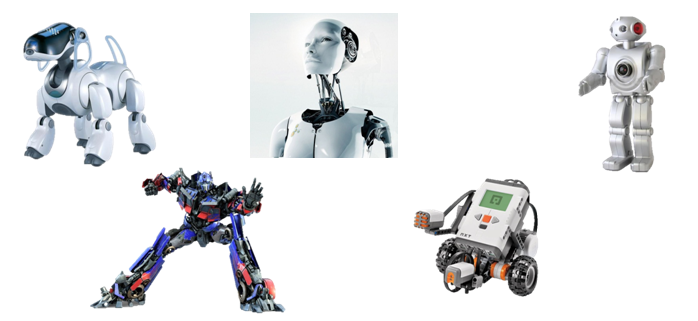
****

图3.1 ROS产品

ROS的一般特征能够总结为下面几条：

（1）点对点设计

ROS系统由一系列进程组成，这些进程是点对点的设计。ROS点对点设计可以降低计算机的计算压力。

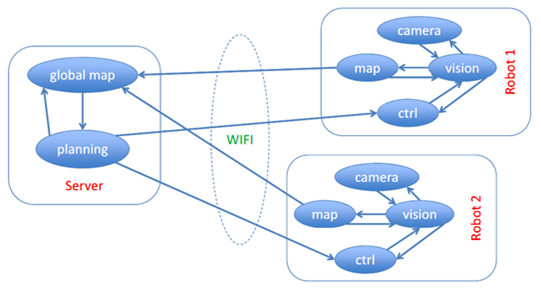


图3.2 点对点设计

（2）多语言支持

在敲程序时，大部分程序员会比较喜欢某种程序语言。因此，开发者设计一个基于ROS框架的多语言支持的系统，如Python，C ++等。



图3.3 多语言支持

例如下面一个消息的接口定义：

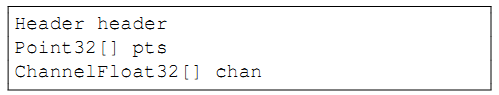


图3.4 消息描述

最终结果是语言独立的消息处理，允许多种语言自由混合和匹配。

（3）精简与集成

ROS运用了很多现有的，成熟的开源的代码库。如OpenCV代码的运动控制和模拟算法。从OpenRAVE的代码的视觉方面。

1. 工具包丰富

ROS包含了较多的适用程序包，能够辅助开发人员快速的开发项目。并且能够减少设计人员或者使用者的工作时间，增高工作的速率。

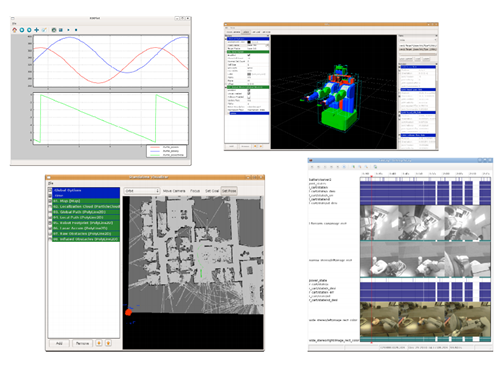


图3.5 运行示例

（5）免费且开源

ROS的全部程序都可以在网上下载到，它是面向全球的开发者和爱好者。任何人，组织机构都可以免费的从网上下载到。

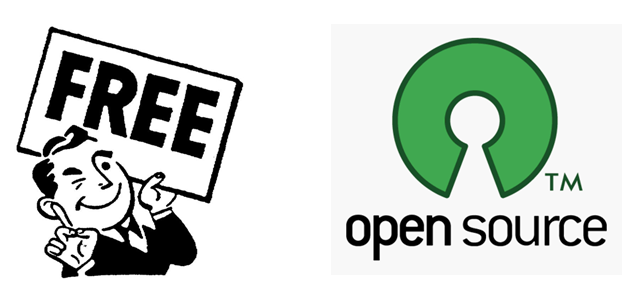


图3.6 ros特点

## 3.3 ROS总体框架

根据ROS代码库的维开发者，一般由两个部分组成：

（1）主要：由Willow Garage计划以及付出主要部分。和许多开发人员提供了一些基本工具，也为整个ROS程序提供了核心。

（2）社区：包括了全球开发者贡献的代码。

通常ROS从另一个视角观察，可以用三个层次来描述：计算机，文件系统级，社区级。

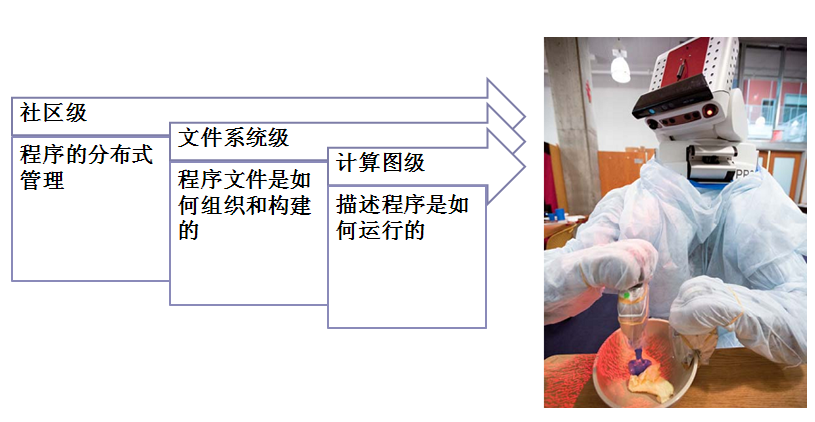


图3.7 总体框架

### 3.3.1 计算图级

计算图是网络上的ROS数据。当软件工作时，其都在做数据整理。 该层次主要包括节点节点，消息，主题，服务（service）等几个重要概念。



图3.8 计算图级

（1）节点

节点是一些直接的算术任务过程。这里，节点能够叫“软件”。当运行更多图像时，咱们应用“节点”使依据ROS的系统。如果较多节点一起工作时，能够较容易地在端对端通信中绘制图表。

（2）消息

节点通过消息进行通信。该音讯能够囊括很多嵌在一起的构造以及数组构造（和C语言的结构有点一样）。

（3）主题

以公布/订阅形式播送的音讯。节点能够公布有关指定内容的音讯。用于涉及特定类型的订阅数据的主题的节点。也许存在节点公布或订阅一样的内容。因此，他们不知道对方的存在。

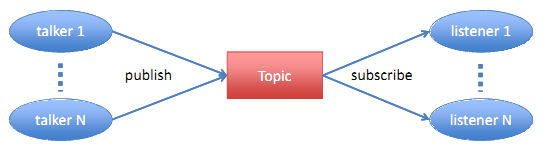
****

图3.9 主题

（4）服务

尽管依据主题的公布/订阅类别是非常有效的沟通方法，然而它公布道路能够容易化一起沟通方法。要留意，和主要内容不一样，有且仅有位移节点能够是泊松服务的独一称号。

ROS提供注册列表和其他计算表查找。没有控制器节点将不能找到对方，替换办事音讯或打电话。

比如，用来管制节点订阅和公布音讯的结构如下：

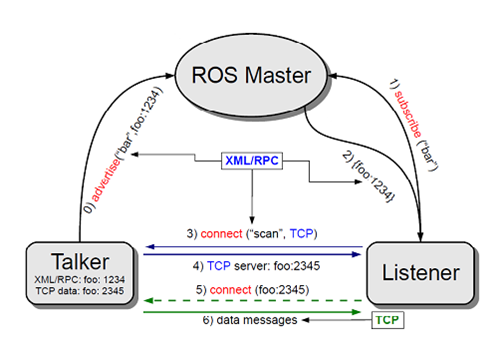


图3.10 服务模型

每个点之间的衔接是直连的，控制的机器能够有信息。订阅主题将需要在出版物和主题之间建立连接，并将根据连接协议建立连接。

### 3.3.2 文件系统级

ROS文件构造一类是指在固态盘上检查的ROS程序的组织。

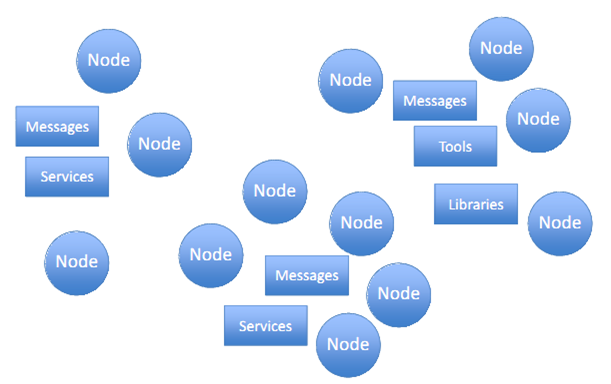


图3.11 文件系统级

ROS中有许多节点，音讯，效劳，设备和程序文件，须要高效的构造来料理程序。 在ROS文件构造一类中，存在一些有用的概念：包，堆栈等。

（1）包：



图3.12 包

ROS软件包来组织。包括了节点，ROS运行依靠的库，适用的数据集合，需要设置的文件或任何其余软件包。逻辑构造是反复应用的目的，采用了易于用的软件来促成结构。通常来说，ROS包短小精干。

（2）堆

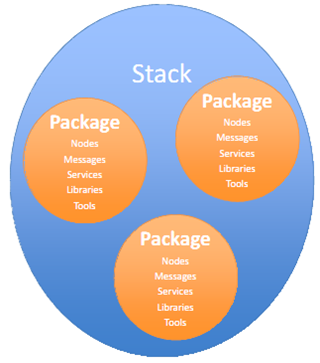


图3.13 堆

堆是供给许多应用的包的集合，例如“导航堆栈”。与版本号相关联的堆栈，也是如何分发ROS软件的关键

### 2.3.3 社区级

ROS的社区档次概念。表现为网络上的代码开释的一种模式。结构如下图所示：

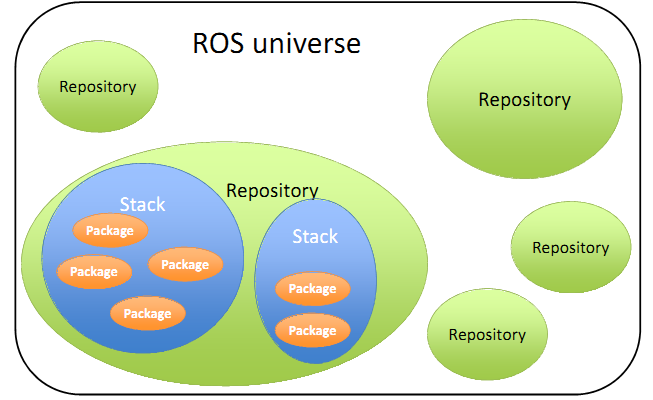


图3.16 社区级

代码库的组合在一起，也那个分开。 从文件系统设计层面到社区层面的协作，可以自主开发和实施工作。 因为这类分开化的构造，使得ROS得到了的极速的进步，尤其是数据包的增多。

# 

# 第4章 目标跟踪技术

## 4.1 视觉目标跟踪概述

4.1.1 视觉跟踪介绍

这里是单个的目的追踪。首帧为手选的框框。在现实中是检测结果，而后追踪算法在以后帧中与当前帧比较。

通常，目标跟踪面临几个主要困难：外观变化，照明变更，极速动作和动作不清，后面景色相似等：

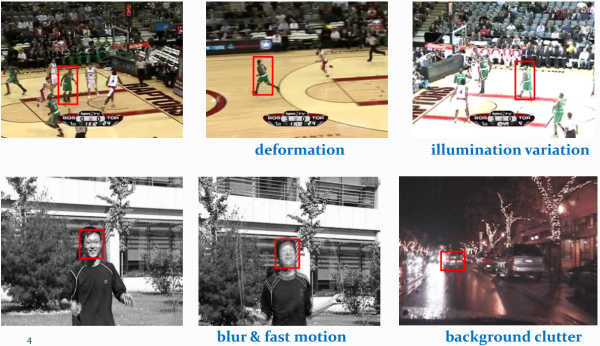
****

图4.1 相似干扰



图4.2 运动干扰

正是因为这些状况，模板跟踪变得艰难，所以要建立健全的跟踪系统，主要考虑的是：鲁棒性，适应性和实时性。

目标追踪程序一般具有下面过程：

（1）获得跟踪目标的适当描述

（2）包含具体的上下文信息

（3）能够自己改变的特征算子。囊括了梯田，色彩，条纹等特征。通常，大部分视觉追踪目的将包含几个步骤，如视频流输入，外观特征表达，周边信息融合和模式的改变。

### 4.1.2 视觉目标跟踪分类

普遍认为有两大类：生成模型法和判别模型法。目前，较为流行的是区分类方法，也称为检测跟踪（跟踪检测）

（1）生成类方法

在目标区域的生成模型里。甚是出名的是卡尔曼滤波。粒子滤波等。比如，目的地区的60％为黄色，40％为蓝色，而去到另一帧中，就类似搜索算法一样去滑动，无处不在找到符合该区域的颜色比例。

（2）判别法

这个措施是图像特征和用铁块的机器去学习。当前的帧到目的地区是最小样本，背景区是虚样本。

分类措施的很近的展开涉及滤波办法，称为CF的关联过滤器，称为DCF的辨别相关过滤器。

他们之间的差异生成判别分类器训练是最大的，在后台信息的过程中使用，所以留意分别器区分前面景色和后面景色，辨别办法常常不错。这不是偶然的，大多数检测算法的每个帧的高复杂度都不能做，这次跟踪较低的计算复杂度是非常合适的，只有在跟踪失败（漂移）或一定间隔后再次进行初始化跟踪器。

## 4.2 核相关的视觉目标跟踪算法（KCF）

### 4.2.1 算法描述

KCF是一种跟踪措施。该办法常常是训练指标检测器。在追踪步骤中，利用指标检测区域语言的下一帧中的目的。而后利用新的测试输出来改变锻炼集。并且更新目标探测器。但是训练目的检查器中抉择正常样本的目的地区时，目的地区四周的不正常样本，比较更靠近目标区域，以取得更多正常样本。

KCF的主要特点：

（1）采取围绕采样周期的正负矩阵。依据线性回归训练目标检测器。矩阵的特征向量。傅立叶空间转圈阵列对角化在Hadamad积的转换中。即点乘法的元素。

（2）经过内核函数映射。线性空间空间非线性脊回归，求解非线性空间和一些罕见束缚的双重效果。

（3）给出了让很多条道的数据融合到程序中的一种办法。

### 4.2.2 线性回归

算法把跟踪成果形象成一个线性回归模型的求解，设代表目的图像的输进来是z,权重为w,输出去是，必要经过训练算得，是目标的特征。令样本最小化为，训练的目的是求最小化平方误差（MSE）下的w：

 （4-1）

其中是希望回归值，是防止特别一致的表达式参数，用于管制结构的构造复杂性，保障分类器的功能，和SVM中类似。其中w可以有以下形式：

 （4-2）

其中矩阵X的每行为，即，I为单位矩阵，。

由于下面的图片要傅里叶域变换，涉及到复杂数的阵列，因此咱们将输出都表示成复杂数的模式：

 （4-3）

其中\*体现复数共轭，H体现转置，即。当X由实数组成时，式（4-2）与（4-3）等价。

方程（4-3）要求逆矩阵，在大规模矩阵运算中，复杂度非常大，计算非常费力，如果X具有特殊的结构，那么可以克服矩阵反演引起的问题，以下将会有效地避免反向操作。

### 4.2.3 循环偏移

其中的示例x能够体现为n\*1的向量，如，咱们可以当做基示例（base sample）,代表第i个示例的特征。本人发热方向是锻炼一个分类器。囊括基样本，和根据移位展现的虚样本。

如果存在换掉的矩阵（permutation matrix）P,它能使基样本产生偏移。

 （4-4）

X乘一次P，X中的元素就向右循环偏移一位，当然如果是负数，也可以向左偏移，由此得到负样本。

循环矩阵如下，表示为。

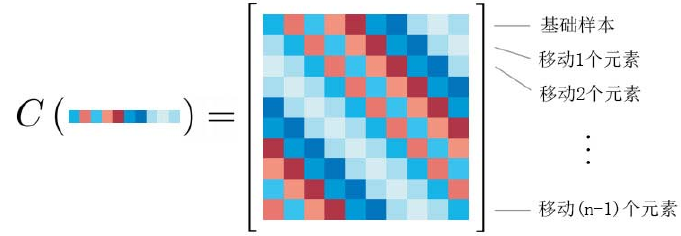


图4.3 循环矩阵示意图

### 4.2.4 循环矩阵

前面4.2.2节末尾说过如果X有特殊的构造，那么就可以克服矩阵求逆带来的困扰，其实这里的特殊构造是指X每一行由基样本x的循环偏移向量组成，如下：

 （4-5）

如果用图像形象化就是： 

图4.4 垂直循环位移示例

当中的基样本图像区分左移和右移要到其他的样本，并且循环矩阵所具有的性质正是帮忙对式（4-3）求解的要害。

循环矩阵的作用：

（1）可以通过离散傅里叶矩阵进行变换，使得矩阵求逆转换为特征值的倒数;

（2）能够转换为频域（4-3）操作，运用散开的傅里叶变化（DFT）进步运转速度。

其中X可以表示为：

 （4-6）

当中是散开的傅里叶矩阵（DFT matrix），表达对x用来散开的傅里叶变动，表示向量对角化。是常量，可以表示为：

 （4-7）

### 4.2.5 整合公式

词节的内容是重要的。根据理论得出一个容易地式子。依据此一个公式能够方便的算出。我们将式（4-6）写入到式（4-3）中（此中也是循环矩阵），成果就出来了。以下是推导过程：





 （4-8）

利用反对角性质：得到：

 （4-9）

再运用圆环矩阵卷积定理：算出：

 （4-10）

### 4.2.6 核技巧（Kernel trick）简要概述

以上做线性回归，可是很多成果都是非线性的。非线性成果往往没有处理，所以咱们能够经过解决线性分类来想出办法。这种求解是一种非线性变换。通过解决原始线性问题。变化后，是非线性问题。这里的非线性变换是使用核技术。

前面几节中的回归问题就是解中的,而式（4-10）给出里解，但是x的线性表示，当运用到实际非线性问题中误差会很大，所以现在需要用x的非线性组合表示：

 （4-11）

假设存在，相等符号左面的为点积，相等符号右面的称为核式子。所以，原线性回归问题就可以表示为：

 （4-12）

### 4.2.7 快速核回归

由上一节可知，最终的线性回归变换为了式（4-12）的非线性回归问题，现在要做的就是求出，为元素的向量。

根据文献【8】，我们可以得到非线性问题的解：

 （4-13）

其中K是n\*n矩阵，。那么又回到了求解方程（4-3）的方法，即如果K是循环矩阵，计算量将大大降低，解决这个问题还得看核函数。

定理1：存在任意偏移矩阵M，都有=，则K为循环矩阵，其中。

满足以上定理的核函数有：高丝核，显性核，很多式核及不少加法核函数。所以可将（4-13）对角化，得到：

 （4-14）

是何惧帧的首行，表示向量的DFT变幻。

关于定义一个更通用的核有关系很有用。任意俩个向量和的核相关，即向量的计算如下：

 （4-15）

它包含了俩个参数的不同相对位移所得到的核相关，所以则是x和它自己在频域的核有关系，我们称作“核自相关”，类似线性情况。

这种类似可以进一步推广，由于一个核等价于高维空间中的点积，式（4-15）的另一种形式是：

 （4-16）

这是和在高维空间的互相关。

注意，我们只需要计算核和运算中的自相关性，这是一个n \* 1向量，采样数与传统的核方法线性增加。相反，传统的计算核矩阵n \* n的方法，采样两次缩放。因为我们知道K的确切结构。你可以比一般的算法更好。

### 4.2.8 快速检测目标

经过以上的措施可以训练为，相当于得到了线性回归方程中的，所以现在可以进行目标检测。

经过建造测量样本和训练样本的核数据矩阵是下面：

 （4-17）

当中是向量，是圆圈矩阵的首行。如此便能够一起计算，按照策士杨笨z的圆圈挪动形成的侧视阳本。它们的输出，即：

 （4-18）

注意，这里的差别于式（4-12），它为向量，由基于基样本z不同循环移动下的输出值构成。依据循环矩阵的性质，变化到DFT域之后，即：

 （4-19）

由此看来，在所有位置估计的值可以看做对核进行空间域的滤波，每个是中相邻核值按照学习到的参数进行的线性组合。

### 4.2.9 快速计算核函数相关性

到现在为止，只剩下突前局部提到的和没有说明如何计算了。

首先列出的计算公式如下：

 （4-20）

其中，表示核数学，是依据x为首行的圆环矩阵，根据循环矩阵的性质，可得：

 （4-21）

其中表示傅里叶变换。

所以，对于多项式核，其计算公式如下：

 （4-22）

关于高斯的核，计算公式如下：

 （4-23）

可以得到，我们能在的复杂度下计算全部的核相关。

### 4.2.10 多通道数据处理

在本文中，能够从灰色特色中提取目的区域的特征，但可以利用HOG特征来取得更好的结尾，那么如何将Hog特征添加到上述模型中。

HOG特色是将部分图片分割成小块，叫做单元格。提取单元格中的突变信息。而后绘制梯度目的直方图。如果要降低光学照明产生的效果，几个单元格方向直方图字符串进行块归一化。细胞直方图最终将在一起是图像的特征。

然后，依据高斯核，扩展到多通道就是下式：

 （4-24）

## 4.3 KCF算法实现

4.3.1 视觉目标检测框架

特征选择和提取

正样本

训练样本

负样本

训练分类器

特征选择和提取

分类结果

分类器

扫描窗口

输入图像

图4.5 目标检测框架

目标检测分为以下几个步骤：

（1）创建培训样本

培训的样本将正，负样本囊括。正例是指要检测的杨奔母表，负示例是如背景一样的图。它们都将标准化到一样大小的图片。

（2）特征提取

由图像或波形取得的数据很大。例如，文本图像可能具有数千个数据，ECG波形也可能具备数千个数据。为了完成分类，一定要开始原始数据的转换，最能表现天然的特色。这是性能分类的抉择和提取进程。一般，最初的数据空间称为测量空间，分开类的依靠的是特征空间。通过变换，可以说在高危区域中特征区域的维数表现较低。

（3）训练分类器

这是第一个了解分类器是什么？我想我能够了解如何，例如：大脑自身就是一个分类器（但它超过想象力，对事物的认可是很大的），这个人本身就是一个分类进程。人们在生长或学习过程中，会经过一些详细的例子来理解一类事物的性质和特色，以查看一类事物。而后在遇到一个新对象后，依据人的特点大脑会事物是依照一类事物的性质和特征。分类为AA类或非AA类。因此，训练分类器能够经过学习的正和负原本理解为分类器（Brain），其具备检测指标的能力。

为了使分类准确率更好，训练样本普通是数万，而后每个样本抽取大批的特色，因而产生了大量的训练数据，所以训练进程一般很耗时的：

（1）使用训练有素的分类器进行目标检测

分类器可以将输入流中的每帧视频图片分类，不管什么目标，都能够在视频流中检测到。检测进程普通是这样的：利用扫描窗口在功夫滑动窗口中检测图像，每个到一个地位，将计算该区域的特色，而后咱们靠训练分类器对特征抉择，确定该地域能否为目的。然后在五月份，因为图像的目标尺寸和分类器的训练您可以对不同大小的图像进行采样。所以咱们可以对子窗口停止更大或更小的扫描，而后在图像中滑动，然后再次般配。

（2）学习和改进分类器

如今假如示例数，特色抉择和份类算法在份类器检测内容比较高的条件下更好。 然而会有虚伪的。所以再多一点先进的话就是增加一个学习或许自适应，也就是说你把这个图片分类的谬误。我拿出这个地图，标识正确的类别，而后给分类器训练样本，让分类器更新，醒来，不要误会我。你怎样晓得他错了？ 我明确，他们大多数都依赖于先前的知识（如目标本身或结构什么约束）或目标和跟踪。

总的来说，这个框架的过程适用于诸如目标跟踪，语音识别等多个领域，那么整个过程的关键点呢？

（1）特征选择：

这些目前更受欢迎：haar特色，lbp特色，hog特色和shif特色等; 它们是不同的，这取决于你想要检测的目的的状况，如：

拳头：纹理特色明显：haar，LBP（现在将结合HOG）;

手掌：轮廓特色分明：HOG特征（行人检测普通应用此）;

（2）分类器算法：

目前的主流意识是：SVM拥护向量机，AdaBoost算法;。行人检测个别是HOG + SVM。OpenCV脸部检测普通是Haar + AdaBoost。

### 4.3.2 软件设计

我们采用C++程序设计语言来实现目标跟踪的功能，并且用到了OpenCV开源计算机视觉库和ROS开源机器人操作系统，在Ubuntu平台下进行软件设计。

下面我们将对参数配置，目标跟踪，模型更新等模块进行阐述和分析。

（1）参数配置

我们通过一个够早函数来进行一般的残书配置,函数接口如下：

KCFTracker::KCFTracker(bool hog, bool fixed\_window, bool multiscale, bool lab)

函数中的参数解释：

hog:我们选取HOG特征作为模型的特征。

fixed\_window: 固定选取窗口的大小

multiscale: 采用多尺度特征

lab: Lab颜色空间

这些参数默认都是true。

函数中的部分默认参数变量：

lambda ： 模型更新参数，默认值是0.0001

padding ：跟踪框的填充倍数，默认值是 2.5，即处理的跟踪区域是在以目标为中心，长宽分别是目标大小的2.5倍的跟踪框。

output\_sigma\_factor：高斯回归目标相对于示例样本的空间带宽，默认值是0.125。

interp\_factor：适应率，即更新模型的记忆因子，hog默认是 0.012，lab默认值是0.005。

sigma：高斯核参数，hog默认是 0.6，lab默认值是0.4。

cell\_size: HOG特征中cell的大小，默认值是4。

（2）跟踪模块

我们在读入一帧图像后，运用KCF算法在另一帧中对目标位置执行推断，具体过程如下：

①首先以上一帧（预测）的目标中心为中心点，提取一个长宽都为目标2.5倍的图像块。（保留：并且和模板大小进行了比较，得到一个尺度因子，得到的结果是较大边长按模板大小赋值，较小边长按尺度因子缩放。）

②提取HOG/Gray特征，下面一节会详细介绍如何提取HOG特征

③对提取到的特征点进行Hanning窗口（余弦）处理。因为循环偏移采样得到的样本在靠近边界处会有一些明显的边界线，导致图像变得不平滑，因此需要乘以一个余弦窗来降低边缘像素的权重。

以上步骤的函数接口是：

cv::Mat KCFTracker::getFeatures(const cv::Mat & image, bool inithann, float scale\_adjust)

④对提取到的不一样的进行快速福利叶变幻，用到的函数接口是：

cv::Mat fftd(cv::Mat img, bool backwards)

⑤计算x（当前）与z（样本）的核相关矩阵，用到的函数接口是：

cv::Mat KCFTracker::gaussianCorrelation(cv::Mat x1, cv::Mat x2)

⑥计算响应值，用到的函数接口是：

real(fftd(complexMultiplication(\_alphaf, fftd(k)), true)

⑦计算响应的机打值和绩效值，并获得其所在地点，用到的函数接口是：

cv::minMaxLoc(res, &pv\_min, &pv, &pi\_min, &pi);

⑧得到当前帧的目标的中心预测值。

该过程的流程图如图4.6所示：

取patch

提取特征

进行Hanning窗口处理

快速FFT变换

给定的模型

高斯核相关

response

求极值

目标预测值

图4.6 跟踪模块流程图

（3）模型更新模块

系统在目标检测过后进入模型更新的阶段，也就是我们所说的训练阶段，为下一帧检测做准备。

模型更新的主要步骤如下：

①首先以上一帧（预测）的目标中心为中心点，提取一个长宽都为目标2.5倍的图像块。

②提取HOG/Gray特征。

③对提取到的特征点进行Hanning窗口（余弦）处理。

④对获得到的特征进行快速福利也变幻。

⑤用x对模型\_tmpl进行更新，按照如下方式：

\_tmpl = (1 - train\_interp\_factor) \* \_tmpl + (train\_interp\_factor)\*x

⑥计算x的高斯核自相关，按照如下方式：

cv::Mat k = gaussianCorrelation(x, x)

⑦计算alphaf,按照如下方式：

cv::Mat alphaf = complexDivision(\_prob, (fftd(k) + lambda));

⑧用alpha对模型\_alphaf进行更新，按照如下方式：

\_alphaf=(1 - train\_interp\_factor) \* \_alphaf + (train\_interp\_factor)\* alphaf;

该过程的流程图如图4.7所示：

取patch

提取特征

进行Hanning窗口处理

FFT

加权更新

tmpl

X

核自相关

k

alphaf

更新

\_alphaf

图4.7 模型更新流程图

# 第5章 系统实现

## 

## 5.1 实际场景运行

首先我们介绍一下启动步骤：

（1）启动ros环境

$roscore

（2）运行ros打开深度相机的节点

$roslaunch openni\_launch openni.launch

（3）启动底盘程序

$roslaunch mrobot\_bringup mrobot.launch

（4）启动追踪程序

$rosrun track\_pkg kcf\_node

（5）程序启动后，在图像窗口内用鼠标左键选取所要跟踪的目标

选定目标之后，小车会通过旋转将目标移动的相机的中心位置。目标在和深度相机相距1.5m时开始追踪，一开始的速度为0.4m/s，速度碎着距离的增大而增加。最大的识别距离是5m。角速度初始为0，碎着目标偏移相机中心的角度增大而增加，最大的角速度为0.75rad/s。

本系统由于kinect摄像头位置较低，所以我们将采用小腿绑目标的方式进行测试，如下图：



图5.1 跟踪目标

目标的选取要尽量避免和背景由较多的重复，最好能有显著的特征，这样小车跟随的时候会有不错的效果。

我们总共测试了四种情况，分别是：前进，后退，左转和右转。下图为测试时的场景：



图5.2 测试场景

## 5.2 系统实验效果分析

在这里我们有个无法解决的缺点，就是在前进的时候，两边的轮子启动时间不一样，这就导致了在刚开始前进的时候小车的方向会偏离一点，不过在启动之后就会运行正常。

注意在测试的时候要固定底盘的重心，不然轮子会打滑。在前进时，小车会在目标距离大于1.5m时开始移动。在后退时，小车会在目标距离小于1m时开始后退，后退速度会随着距离的减小而降低，所以此时要小心不能碰到小车。左转和右转情况基本相同，在目标转过较大角度时，小车旋转的非常快，所以要把上面的东西固定好。

为了可以达到更好的效果，我们可以更换一对摩擦力较大的轮子，或者在比较粗糙的地面进行测试，但是不能有过陡的障碍，因为电机的扭矩较小，不能越过比较陡的障碍。

# 结 论

本系统在设计过程中，从最初的方案制定一直到选定方案，真的废了好大劲才最终选定用KCF的目标跟踪方法。在这个过程中，我查阅了大量的资料，从一开始的“懵懂”到最后的“成熟”，历尽了千难万险，我很庆幸自己没有放弃，始终相信自己能够完成。在这个设计中，我认为最难的部分就是目标跟踪算法的实现，我一开始参照的是，基于梦特卡洛方法的粒子滤波，但是，又找到了追踪成果相对来说不错的，基于核相关的滤波算法，他们俩个相比，我觉得后者更易理解和实现。好在我找到了原作者写的论文，然后接下来就开始一步步去啃英文论文，在此期间瞬间感觉自己的英文水平更进了一步。总的来说，系统最终实现了，效果也是不错的。唯一的遗憾就是硬件和机械结构不太让我满意，导致我花费了比较的多时间去调试硬件。尽管我不是个完美的人，但我一直希望大学的最后一个作品能够尽量“完美”，虽然希望落空了吧，但重要的是真的在这个过程学到了好多的知识，让我对自己也有了一个重新的认知。

# 参考文献

1. 刘志强.基于核相关滤波的高速目标跟踪算法研究与系统实现[D].西安电子 技大学硕士学位论文.2015.
2. 赵璐璐.基于相关滤波的目标跟踪算法研究[D].北方工业大学硕士学位论 文.2013.
3. 罗雅愉.基于核相关滤波的目标跟踪算法研究[D].华南理工大学硕士学位论 文.2015.
4. Henriques, João F, et al. "High-Speed Tracking with Kernelized Correlation Filters. " High Performance Computing, Networking, Storage and Analysis, SC14: International Conference for IEEE, 2014:375-386.
5. 陈东成.基于机器学习的目标跟踪技术研究[D].中国科学院大学博士学位论 文.2015.
6. 贺超.基于kinect的移动机器人同时目标跟踪与壁障[D].太原理工大学硕士 学位论文.2010.
7. 朱志宇.粒子滤波算法及其应用[M].科学出版社.2010.
8. 于仕琪，刘瑞祯.学习OpenCV（中文版）[M].清华大学出版社.2009.
9. 刘锦涛，张瑞雷.ROS机器人程序设计（原书第2版）[M].机械工业出版 社.2016.
10. 高翔，张涛.视觉SLAM十四讲-从理论到实践[M].电子工业出版社.2017.
11. Boddeti, Vishnu Naresh, T. Kanade, and B. V. K. V. Kumar. "Correlation Filters for Object Alignment." IEEE Conference on Computer Vision and Pattern Recognition IEEE Computer Society, 2013:2291-2298.
12. Kalal Z, Mikolajczyk K, Matas J. Tracking-Learning-Detection.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2012, 34(7):1409.

# 

# 致 谢

在我整个本科的学习生活中，我时刻能感受到指导老师和实验室同学对我的帮助和理解，以及舍友对我的容忍。在此，我想他们表达最真诚的感谢。

首先感谢我的指导导师岳洪伟老师，在此期间，岳老师对我的工程实践给予了充分的指导和肯定，使得我的综合能力能够快速提高。他一丝不苟教诲和严谨的治学态度都使得我终身受益。

感谢实验室同学对我的学术指导和帮助，有好多次都是他们的答疑解惑才使得我能够搞懂某个知识点，帮助我加快了项目进度，使我能够如期完成项目设计。他们每个夜晚的陪伴也给了我很大的鼓舞，是自己感觉不是一个人在战斗。

感谢舍友对我的“不杀之恩”，以及他们对我的陪伴和鼓励。

特别感谢我的家人和父母，感谢他们对我的支持和理解，他们的关心是我前进路上的精神支柱，因为有他们的支持我才能度过人生道路上的重重困难，不断地前行。

再次由衷的感谢给予我帮助的老师，家人以及同学们！