

# 基于立体视觉与末端摄像头的 机械臂准确抓取系统

清华大学第三十四届“挑战杯”  
学生课外学术科技作品竞赛正式文档

## 摘要

家庭机器人是机器人发展的一大方向,但目前主流的家庭机器人无法抓取物品,更难以进行生活中其他常见的对物体的操作。本文从机械臂出发,提出并设计实现了一套适用于室内环境的抓取系统,结合立体视觉与末端摄像头,使用针对室内场景优化的视觉算法定位物体、识别物体,并引导机械臂进行准确的抓取。另外,为了增加机械臂的使用范围,我们为机械臂增加升降台与移动底盘,让机械臂能够在室内自由的移动,取放更大范围内的物体。本项目经过大量实验,验证了抓取系统在实际生活场景中的适应能力。

**关键字:**视觉伺服,物体识别,立体视觉

# 目录

<b>1 项目背景</b>	<b>3</b>
<b>2 项目创新点</b>	<b>3</b>
2.1 使用末端视觉进行视觉伺服 . . . . .	4
2.2 物体分辨与识别 . . . . .	4
2.3 灰度熵分离前后景 . . . . .	4
<b>3 项目内容</b>	<b>5</b>
3.1 整体架构 . . . . .	5
3.1.1 系统整体架构 . . . . .	5
3.1.2 通信架构设计思路 . . . . .	5
3.1.3 视觉系统设计思路 . . . . .	6
3.1.4 坐标定义 . . . . .	6
3.2 项目设计及实现细节 . . . . .	6
3.2.1 图像处理 . . . . .	6
3.2.2 深度视觉 . . . . .	9
3.2.3 末端视觉 . . . . .	11
3.2.4 机械臂设计 . . . . .	11
3.2.5 机械臂控制 . . . . .	15
3.2.6 距离传感器 . . . . .	16
3.2.7 移动底盘 . . . . .	16
<b>4 项目成果</b>	<b>18</b>
4.1 实物图片 . . . . .	18
4.2 测试结果 . . . . .	18
<b>5 发展前景与应用价值</b>	<b>20</b>
<b>6 总结</b>	<b>20</b>
6.1 项目进展 . . . . .	20
6.2 展望 . . . . .	21

## 1 项目背景

随着智能化控制的进步,工业4.0时代逐渐到来,智能制造与智能家居产业发展迅速,机器人也慢慢走近每家每户。许多家庭中配备了扫地机器人用来自动清扫地面,让人在每天忙碌的工作后不必再面对扫地的烦恼。智能家庭机器人Jibo能够识别人的情感并与人产生交互,为家庭生活平添了一份乐趣。而在未来,智能家居机器人可能成为家庭的一员,完成更加复杂的任务,辅助人们日常工作、学习、娱乐,替代人类完成繁杂、琐碎的家务活。



图 1: Dyson 扫地机器人



图 2: Jibo 家庭机器人

目前,市场上大多数的家庭机器人还都停留在小型的阶段,虽然在软件上已经达到了能感知环境、理解情感、流畅交流的程度,但仍然无法摆脱自身硬件方面的限制,不能帮助完成许多日常生活中看似简单的任务。我们在日常生活中最基本的操作就是拿取物品,而小型的机器人由于体型过小或没有可移动部分,无法做出这些动作,更不可能做出把一个指定的东西送到手边、整理桌上杂乱物品这类更加复杂的动作。另外,目前的智能电器大多只是一个个分立的器件,无法构成一个整体,给人造成了许多的不方便。比如,智能洗衣机在洗完衣服后仍然需要人去晾衣服,智能冰箱也需要人去开门拿食材。这些“最后一步”的问题也让智能家居的体验大打折扣。

因此,我们希望实现一个可以移动的机械臂,能够在日常的生活场景中自动识别并抓取、放置物体,连接人与物品,连接人与家居。它能从摆放了各种物品的柜子上帮你取下指定的那一个,或者把桌上杂乱的物体排列整齐。基本的物体操纵功能在实际生活中使用十分广泛,我们的机械臂也将胜任日常生活中的大部分任务,能帮人大大减轻负担。

## 2 项目创新点

在复杂多变的现实场景中,机械臂自动抓取的任务较为困难,但目前已经有一些研究成果。瑞士洛桑联邦理工学院的 LASA 实验室使用能获取关节角度信息的轻量级机械臂 KUKA, 经过手动训练, 通过外部相机的视觉信息能准确抓取飞行中的物体 [1]。卡内基梅隆大学 NREC 部门的 ARM-S 机器人利用三维传感器获得场景点云并提取识别物体, 采用关节反馈来准确获得机械臂的位置 [2]。这些机械臂都内置了关节反馈, 对电机精度也提出了较高的要求, 导致机械臂制造成本较高。另外, 由于缺少机械臂的末端视觉反馈, 机械臂在移动时使用开环控制, 难以应对现实场景中突发的变化。

为此, 我们提出了一套抓取系统, 利用灰度熵与准对室内场景优化的算法来分离物

体,使用基于稀疏表示的分类器来识别物体,采用末端视觉反馈来引导机械臂的抓取。这套结合了多种算法的系统能够很好地完成室内环境的物品抓取任务。具体创新点如下。

## 2.1 使用末端视觉进行视觉伺服

我们的机械臂最基础的功能就是抓取物体,而在复杂的室内环境中,通过深度视觉获取到的物体位置信息不够准确,加上机械臂本身的旷量与电机角度的误差,若用开环控制机械臂的移动,很容易导致抓取的不准确,还有可能让机械手触碰到附近的其他物体。

为了解决这一问题,我们在机械手上加装了末端摄像头,使用基于图像的视觉伺服(**Image-Based Visual Servoing**),构成一个手眼(**eye-in-hand**)机械臂系统,通过末端视觉获得的物体位置作为反馈,构成机械臂移动的闭环控制,保证物体的准确抓取。

## 2.2 物体分辨与识别

物体识别的一般方法是通过计算两个矩阵的标准化降采样矩阵的点击来判断相似性 [3],这样做虽然有一定的效果,但结果的点积值却没有实际意义,对于不同的物体相差非常大,因此难以取合适的阈值,判断的鲁棒性不高。

我们采用了基于稀疏表示的分类器来解决这一问题。首先,我们假设当样本数量足够时,每张该物体的照片都可以近似表示为该物体样本照片的线性叠加。随后,使用稳定 $L_1$ 范数最小化问题估算稀疏解,通过残差来确定识别结果。使用这种方式,算法的鲁棒性有了显著的提升,同时速度和存储空间利用效率上并没有显著的降低。

## 2.3 灰度熵分离前后景

在项目的实践过程中,我们需要一种方法,实时地在以书架为背景的典型场景中,找到所有目标物体的三维坐标。已知的目标查找方法有很多,但鉴于环境的复杂性和图片的质量,已知的算法很难在很短的时间内完成标记目标的任务。

我们针对书架场景的特点提出了基于熵过滤的目标查找算法。首先使用灰度熵的方法过滤背景,但对于纯色物体与书架边框区域会产生误判。作为弥补,我们又提出了目标纯色快提取方法和线段检测方法,分别查找纯色物体和过滤书架框架。利用过滤结果,配合 Kinect 获取得到的深度信息,再配合三维墙面提取、欧几里得聚类,最终可以确定每一个物体的三维坐标。

### 3 项目内容

#### 3.1 整体架构

##### 3.1.1 系统整体架构

系统的主要由以下部分构成：

1. 运行于主计算机 ROS(机器人操作系统) 上的算法与决策层
2. 运行于 Xilinx ZYNQ 7020 芯片上的 ROS 系统的硬件接口层, 用于对主计算机难以直接控制的硬件进行信号采集和驱动
3. 深度摄像头 Kinect, 用于初步的物体识别
4. 机械臂, 用于物体抓取
5. 机械臂末端摄像头与超声波传感器, 用于末端的目标位置修正与反馈
6. 机器人移动平台, 用于扩展机械臂的抓取范围

各个部分之间的连接方式如图3所示。

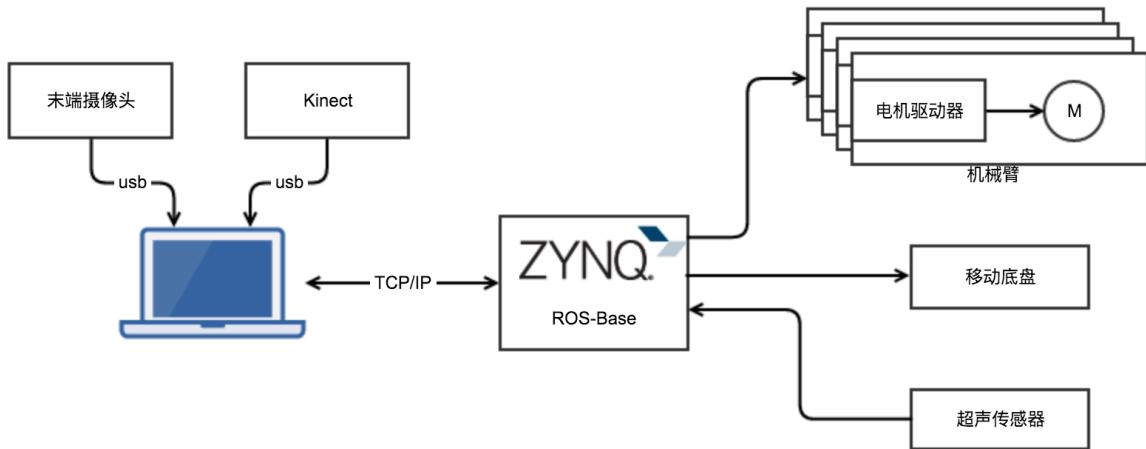


图 3: 整体系统架构

##### 3.1.2 通信架构设计思路

系统整体的硬件通信架构基于机器人操作系统 ROS 构建。ROS 是一个基于网络协议的通信系统, 支持分布式的节点。在主机器人上运行的节点负责图像处理与机械臂的运动规划。规划的结果通过网络发送到同样运行着 ROS 的 ZYNQ 芯片上。ZYNQ 芯片是一种同时集成了 ARM 微处理器和 FPGA 可编程逻辑的全可编程器件。

在 FPGA 上构建了用于直接控制电机并且采集超声波传感器反馈的外设。而在微处理器上运行的 ROS 节点通过网络从主计算机运行的 ROS 节点上收到控制机械臂姿态目标,转换为实际电机的 PWM 脉宽写入电机控制外设。同时另一个 ROS 节点采集超声反馈信息通过网络发送到主计算机上用于之后的运动规划。

### 3.1.3 视觉系统设计思路

Kinect 深度摄像头可以给出 0.6m 到 4.5m 之内的立体图像。通过图像处理方法分辨出深度图像中的物体。然而由于 Kinect 深度摄像头本身的分辨率和净度所限,会产生深度和位置信息不准确以及误报物体等情况。

为了修正这些误差,我们使用了一个机械臂末端的摄像头。主计算机从末端摄像头图像中提取出可能的物体区域,和预先采集的物品图像库进行匹配。如果匹配失败,就认为该物品为 Kinect 误报。如果匹配成功就开始物品的抓取,在抓取过程中利用末端摄像头看到的物品图像进行反馈。反馈目标为将物品图像定位到末端摄像头图像的中心,同时通过超声波传感器信息采集机械手到物体的距离就可以达到准确抓取的效果。

实际抓取过程的详细流程图如图4所示。

### 3.1.4 坐标定义

在整个抓取过程中使用的坐标定义如下:

1. 机器人本体坐标系 `base_link`, 以机器人正前方为 x 轴, 正左方为 y 轴的右手系
2. Kinect 坐标系 `kinect_link`, 以 `kinect` 正左方为 x 轴, 正上方为 y 轴的右手系
3. 末端视觉坐标系 `hand_link`, 以机械手正前方为 x 轴, 正左方为 y 轴的右手系

## 3.2 项目设计及实现细节

### 3.2.1 图像处理

**灰度熵** 以图5为例,我们将演示分离书架上的物体和书架的算法。直接利用灰度熵,去除书架背景,结果如图6所示。对于一张给定的图片,其熵的定义如下:

$$E = -\sum p_i \cdot \log_2 p_i \quad (1)$$

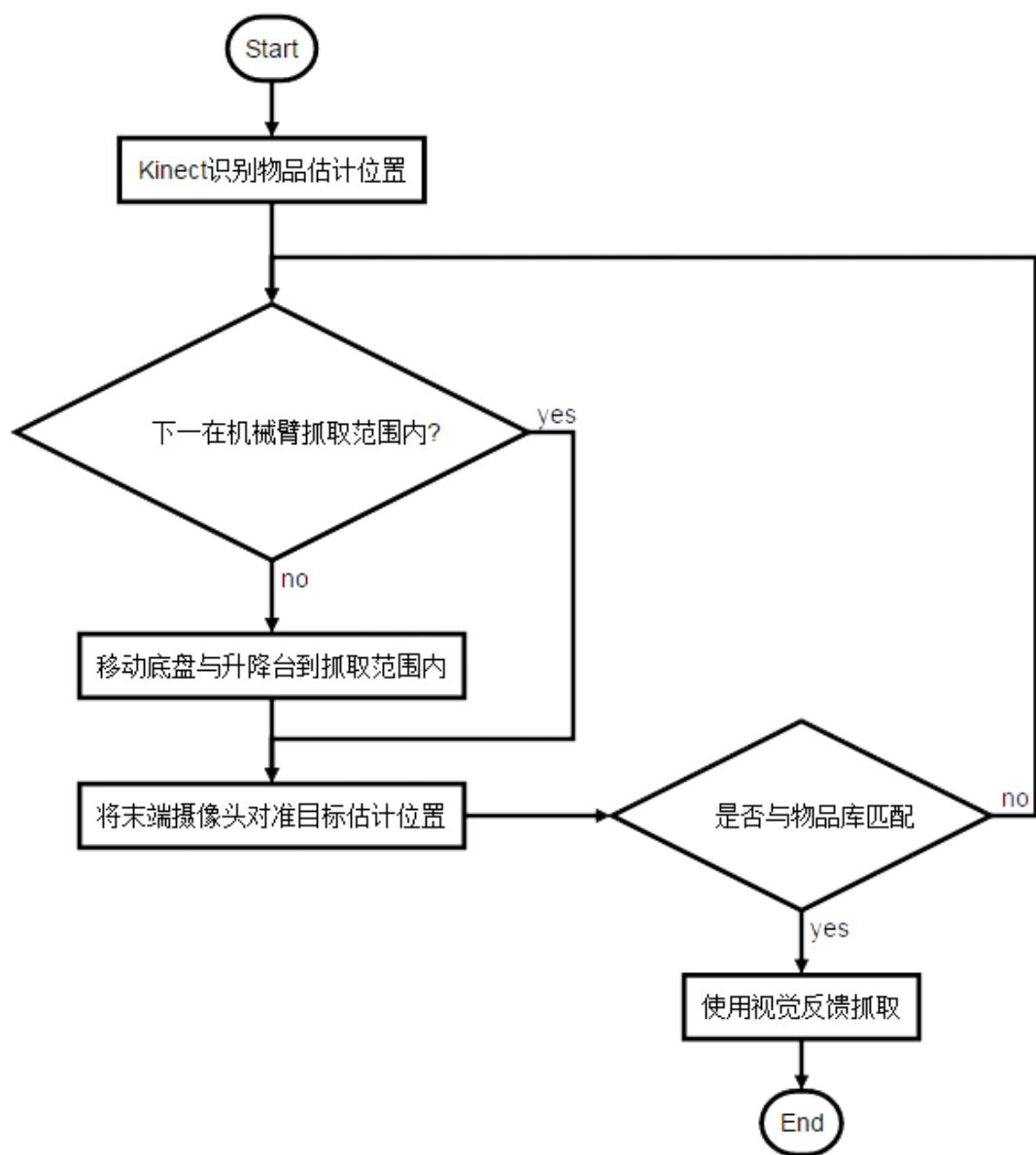


图 4：抓取流程图



图 5: 示例图片



图 6: 灰度熵结果

在这张图片中,书架被过滤得很干净。不过那些纯色的物体也被过滤了。此外,书架的框架也没有去除干净,所以这张图片还不能用于目标的定位和识别。针对熵过滤的不足,我们又提出了色块检测、线段检测方法。

**色块检测** 在图片进行边缘检测之后,降采样,找到其中的纯色块。其中位于边缘部分的色块意义不大,直接忽略,结果如图7。



图 7: 色块检测结果



图 8: 色块分离结果

这些色块中,既有背景色块,又有目标色块,需要将他们分离。首先,将剩余色块中面积最大的色块视为标准背景色块(如果标准背景色块的面积比较小,则认为所有色块均为目标色块,终止背景色块识别操作)。检查所有色块,面积同标准背景色块相近、或者颜色同标准背景色块类似的色块,均被视为背景色块。其余的部分直接视作目标色块。最后,将降采样的结果重新应用到图片上,获得图片的色块层,如图8。

可以观察到色块检测也存在缺陷,在背景颜色起伏较大的细节区域可能无法准确辨别背景色块和目标色块。这些区域将会在平面检测时过滤。

**线段检测** 在图片进行边缘检测之后,降采样,如图9所示。将边缘点过分密集的区域清理掉,这些特征点密集的区域应当是目标,结果如图10。



图 9: 边缘检测结果



图 10: 边缘区域清理结果

实践时发现,纯色块的边缘也可能被意外地检测为线段,所以将纯色块区域附近的边缘删除,如图11。最后,使用 Hough 检测其中的线段,结果如图12。



图 11: 纯色快边缘删除结果

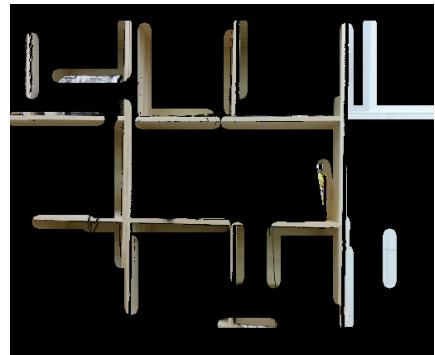


图 12: 线段检测结果

**总结** 最后,我们可以合成得到目标图层:



图 13: 最终结果

### 3.2.2 深度视觉

利用 Kinect2 我们可以获取场景的 RGB 图像,以及 RGB 图像中每一个点对应的到镜头的距离。我们完成对 Kinect2 设备的标定后,就获得了深度信息到机器人坐标系的变换矩阵,这样每一个对应的二维图层都可以转化为与之对应的三维点云模型。

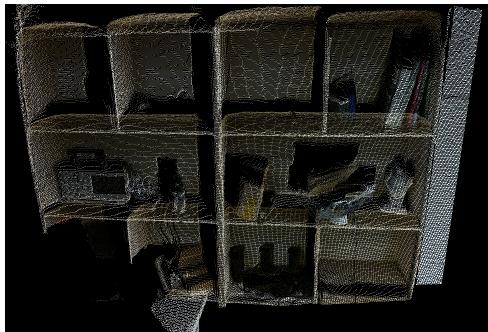


图 14: 原始点云



图 15: 确定的背景

为了进一步清理 RGB 过滤中获取的信息, 实现目标的准确定位, 我们将利用目标图层和书架背景图层对应的三维模型进行深度信息的处理。我们在书架背景图层中找到一个最大的平面作为墙面, 以墙面的坐标为基准, 去除目标图层中在那些在墙面上的点。最后, 我们对清理后的背景图层进行欧几里得聚类, 将每一组几何关系上相互连接的点认作是一个目标, 完成全部目标的定位工作。



图 16: 物体图层

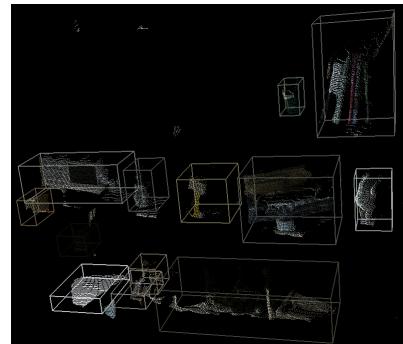


图 17: 聚类后的物体图层

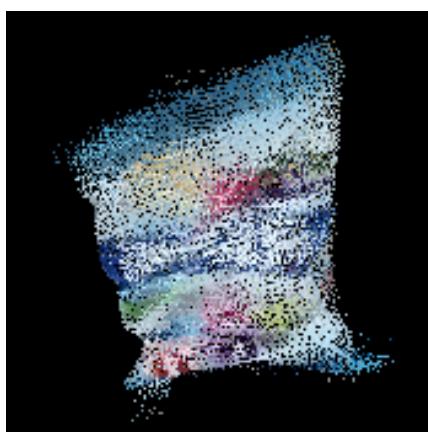


图 18: 糖果袋



图 19: Baymax 模型

图 20: 物体点云

### 3.2.3 末端视觉

我们从末端摄像头获得原始图像,其分辨率为  $640 \times 480$ ,以  $10\text{Hz}$  的频率从摄像头读取,并且包装成 `ros` 信息后被发布,供后续模块读取。正如上一部分所说,对于室内场景的图片,我们可以使用灰度熵来处理图像,对于末端视觉我们也采用了这种方法。处理之后我们便获得了物体的图像,接着我们需要辨别物体的类型。作为前提,我们不妨直接取所有物体中轮廓最大的那个作为研究的目标物体。

**物体识别** 我们使用基于稀疏表示的分类器 (**Sparse Representation-based Classification**) 对上述的图像进行识别。首先,我们事先准备该物体一定数量的样本,并假设当样本数量足够时,每张该物体的照片都可以近似表示为该物体样本照片的线性叠加,并且与其他物体的样本照片线性无关,这也就是“稀疏表示”的含义。数学形式的表示为:

$$\mathbf{y} = A \mathbf{x}_0 \quad (2)$$

其中  $\mathbf{x}_0 = [0, \dots, 0, a_{i,1}, \dots, a_{i,n_i}, 0, \dots, 0]$ , 若该物体属于第  $i$  类。

可以证明 [4], 稀疏解  $\mathbf{x}_0$  可以近似的被如下的表达式所估算:

$$\hat{\mathbf{x}}_1 = \arg \min \|\mathbf{x}\|_1 \quad \text{s.t. } \|A\mathbf{x} - \mathbf{y}\|_2 \leq \varepsilon \quad (3)$$

并且可以证明,  $\hat{\mathbf{x}}_1$  能充分接近  $\mathbf{x}_0$ 。具体的实现方法如下:

1. 首先给定一系列样本矩阵  $\mathbf{A}$ , 被测试样例  $\mathbf{y}$ , 以及可容忍误差  $\varepsilon$ ;
2. 将  $\mathbf{A}$  标准化使其有单位的  $L_2$  范数, 然后求解方程(3);
3. 依次计算残差  $r_i(\mathbf{y}) = \|\mathbf{y} - A\delta_i(\hat{\mathbf{x}}_1)\|_2$ , 其最小值即为我们识别的结果。

**降采样** 为了降低  $\mathbf{y}$  的维数, 从而加快运算速度, 对图片进行特征提取是很有必要的。从数学上可以证明, 只要提取的特征向量的维度大于  $O(\log n)$ , 就不会影响方程(3)的解, 因此我们可以直接对图片进行降采样。

### 3.2.4 机械臂设计

**结构设计思路** 选用的舵机能够负载  $30\text{N}\cdot\text{m}$ , 舵机行程共  $270^\circ$  左右, 动力充足, 行程宽, 故希望能够将舵机的有效行程缩短至所需的范围, 以此优化机械臂的动力分配性能。

**第一版** 如图21所示, 第一版机械臂为兼容延续之前的机械设计以及成果, 采用了较为灵活的钣金结构设计, 沿用了前作的底座, 有效的节省了设计周期与成本。

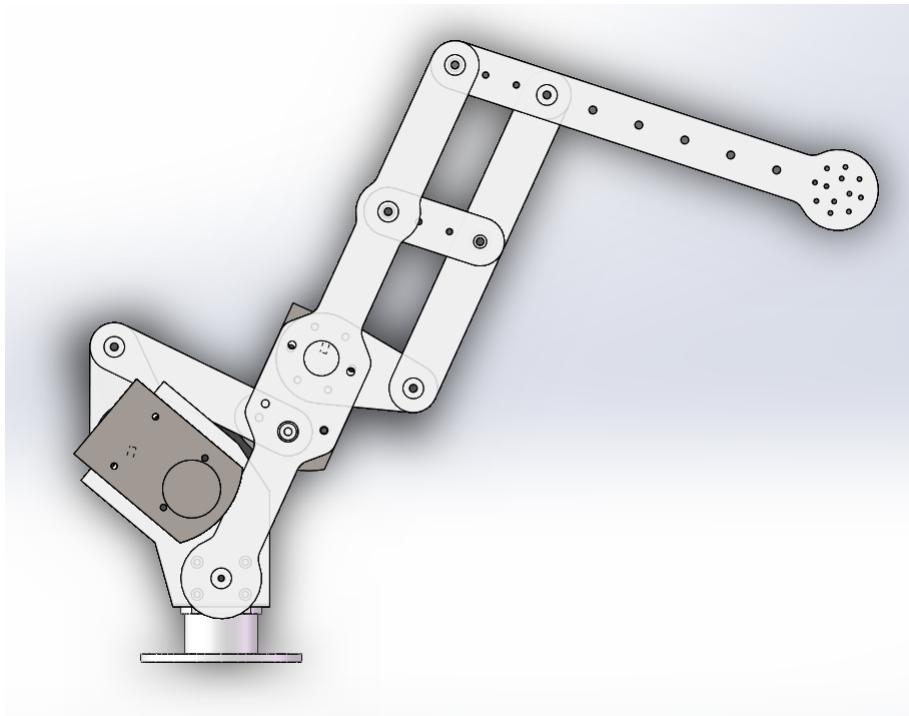


图 21: 第一版机械臂外观

在图22中可以看到有关机械臂的设计的原始尺寸, 其主要目的为规划舵机的行程, 使得行程充分转化, 以达到优化的目的。

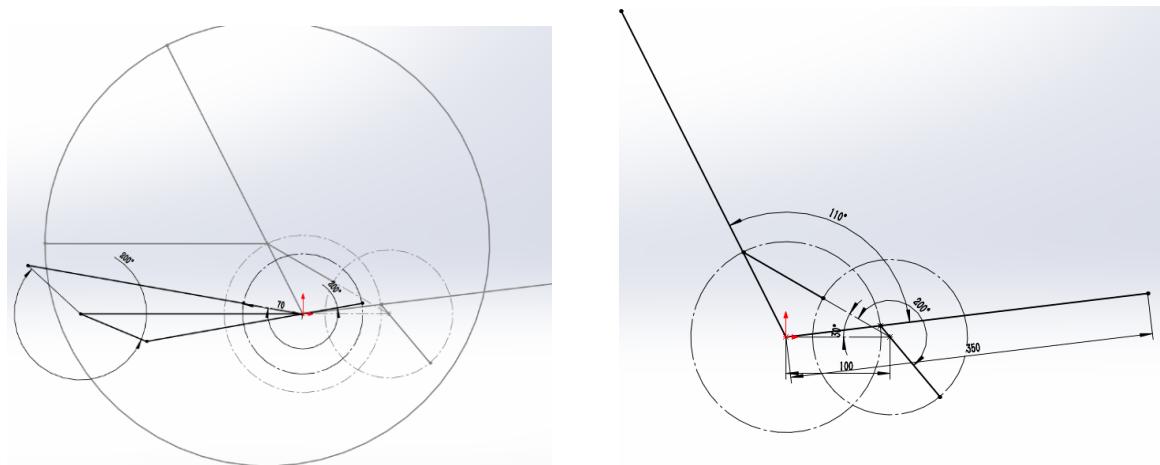


图 22: 机械臂行程规划图

如图23所示, 第一版机械臂的载荷(蓝色)随机械臂上提而减小, 同时机械臂的动力却在逐渐上升, 从而在运动的中后段能够灵活有效的运动, 但在前段会出现一段较大的动力不足的区间。

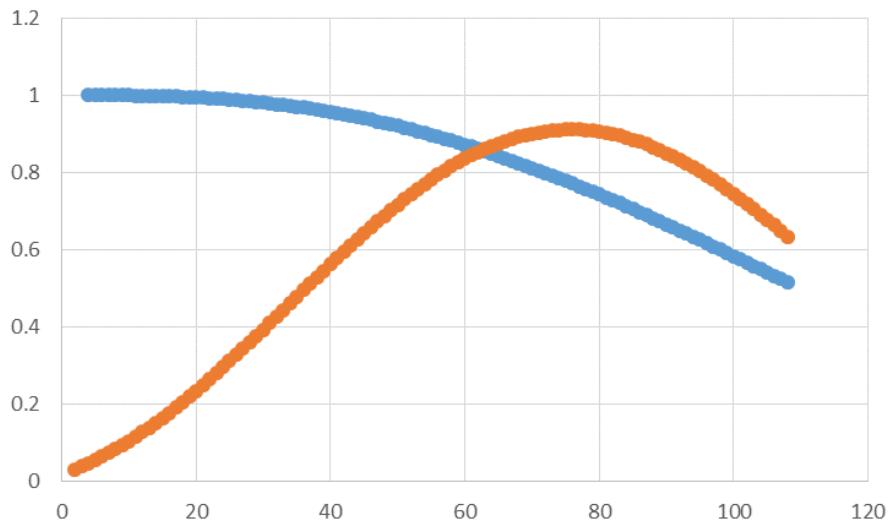


图 23: 第一版机械臂负载与载荷关系

第二版:如图24所示,第二版在前一版的基础之上加装了一个自由度,同时增加了许多的横向加固的内梁结构,能够有效的解决前一版机械臂抗扭曲刚度不足的问题。

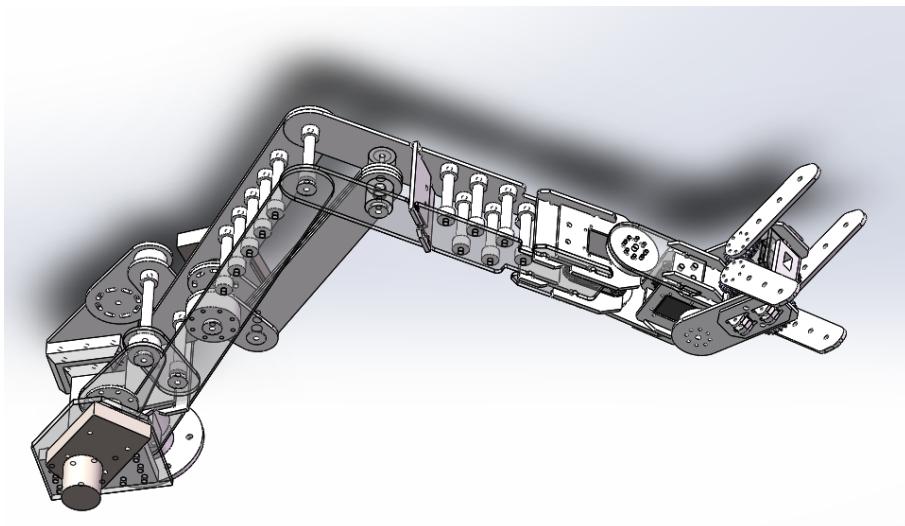


图 24: 第二版机械臂总成图

如图25所示,在第二版的机械臂设计中,吸取了上一版的机械臂的长处与优点,并且针对前一半的机械臂在前半段的运动区间内有较大的动力不足的情况作出了改进,改进后的载荷与负载关系如图26所示。

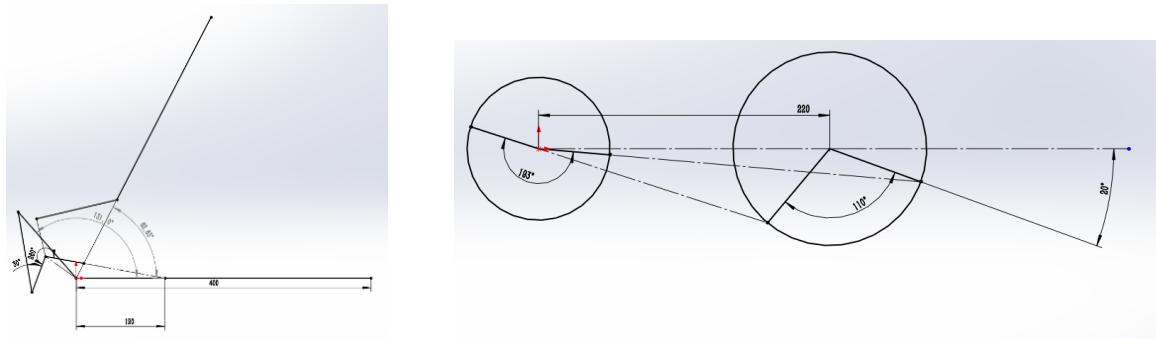


图 25: 机械臂行程规划图

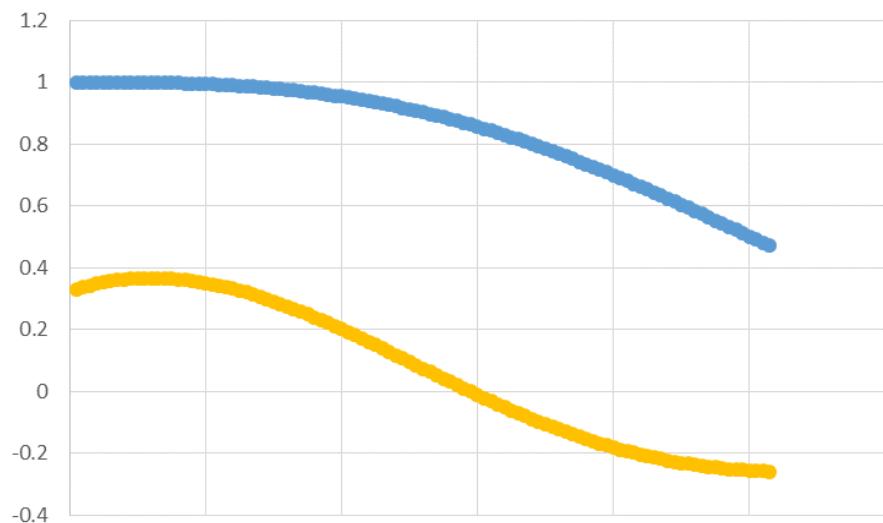


图 26: 第二版机械臂动力分配效果

**其他创新设计** 在第二版的机械臂设计中,拟增加自适应机械手环节,使得  $70\text{N}\cdot\text{m}$  的舵机能够发挥充足的握力,拥有臂上一版更加优良的抓持稳定性,其自适应手结构目前仍在设计阶段,初期尺寸规划及行程安排如图27。

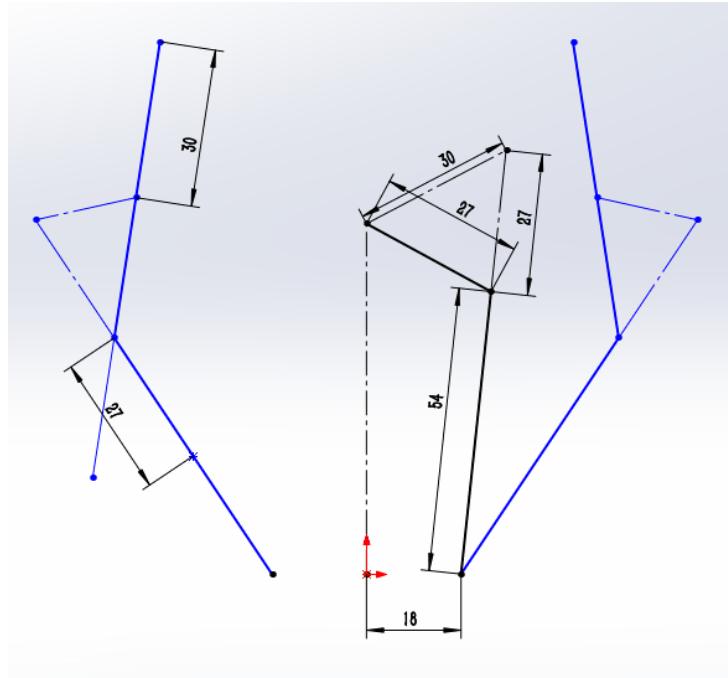


图 27: 自适应手关节初期设计草图

### 3.2.5 机械臂控制

考虑到控制算法的可扩展性, 我们采用了通用的机械臂位姿算法。首先, 建立机械臂对象, 输入机械臂每一段的转轴方向、相对于下一个关节的位移即可。在建立了模型后, 便可通过机械手的目标位置解算出电机的目标角度。求解时采用了一种非线性优化算法, LM 算法 (Levenberg-Marquardt)[5]。它将高斯-牛顿法与梯度下降法相结合, 从初始点开始迭代, 通过合适的阻尼因子  $\lambda$  来达到最终解。这个算法能够有效处理冗余参数的问题, 在处理实际问题时效果良好。

在具体的实现过程中, 由于算法对初值敏感, 所以我们会在机械臂当前角度作为初值求解失败后采用随机的角度作为初值重复求解直至得到结果。

在每一次抓取物体之前, 机械臂会回到初始位置, 从深度视觉获得目标物体的三维位置信息作为机械手的初始目标。机械手向前移动, 在此过程中不断接收末端摄像头识别出的物体位置, 依此在  $yz$  平面上调整目标位置, 形成闭环控制。距离传感器不断探测机械手与物体的距离, 在到达阈值后让机械手上的电机转动, 抓取物体。

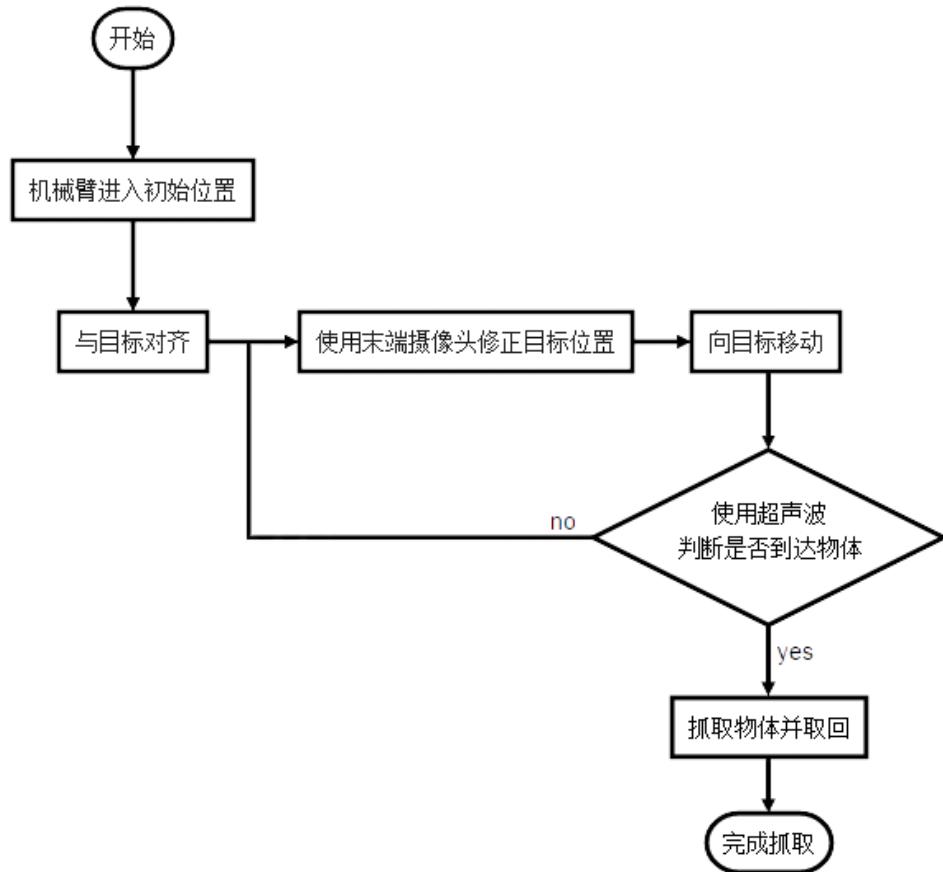


图 28: 机械臂流程图

另外,为了保证深度视觉信息的准确,在 Kinect2 获取场景图像前,机械臂将旋转至一个边缘位置,到 Kinect2 的视野之外,减小对于图像与深度信息的干扰。

### 3.2.6 距离传感器

由于单目的末端摄像头无法准确获得物体的深度信息,通过深度视觉获取的深度信息精度也无法保证,所以我们在机械手上加装了末端超声波传感器,用来准确探测与目标物体之间的距离。在距离达到阈值后,便告知机械臂已到达目标位置,可以开始抓取物体。

### 3.2.7 移动底盘

在底盘设计中,原本采用四个独立正交的欧姆尼轮作为运动输出,但在实际使用过程中发现该设计的通过性较弱,不能够适应复杂的环境,容易被线缆等物体牵制。故在后续的设计中,我们采用麦克纳姆轮作为动力输出,使得底盘能够在保持全向运动性的同时获得更大的带载能力与更好的通过性。

同时,在新的底盘设计设计中,采用了 6063 铝型材作为底盘的主框架,有效的增加了底盘的可扩展性与比强度,简化了设计的难度。

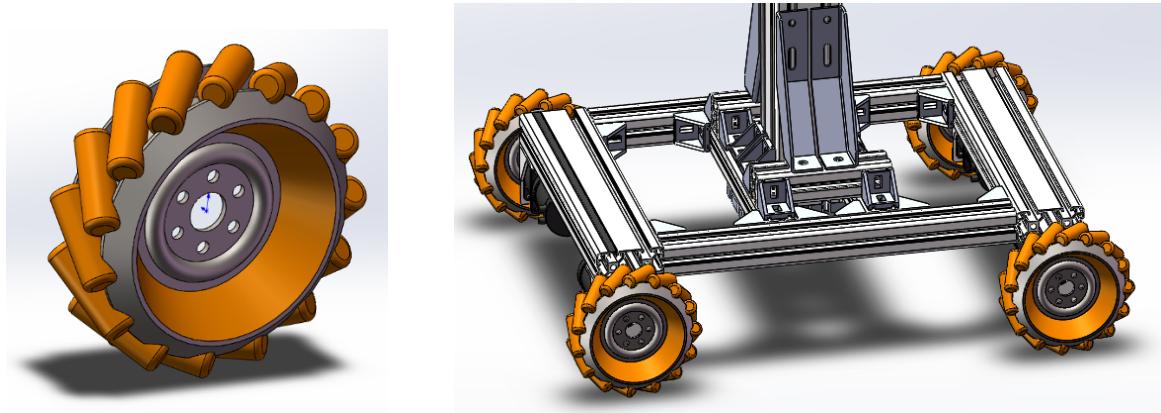


图 29: 麦克纳姆轮及底盘

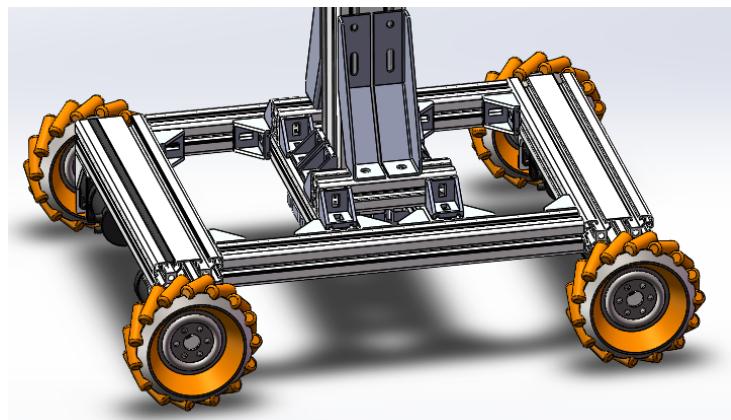


图 30: 选用的欧标 6063 铝型材横截面示意图

## 4 项目成果

### 4.1 实物图片

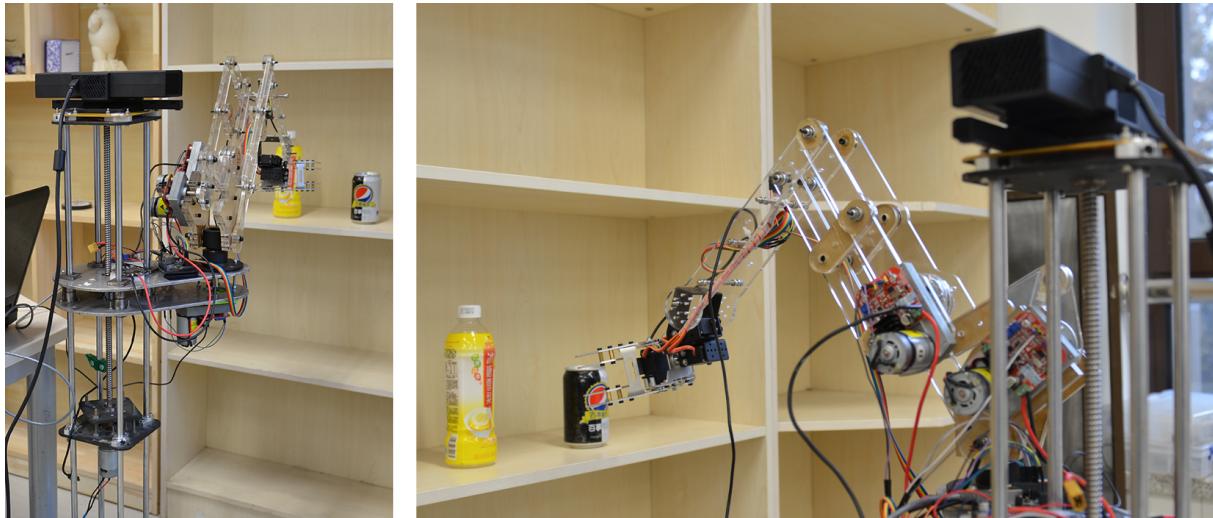


图 31: 第一版机械臂实物图

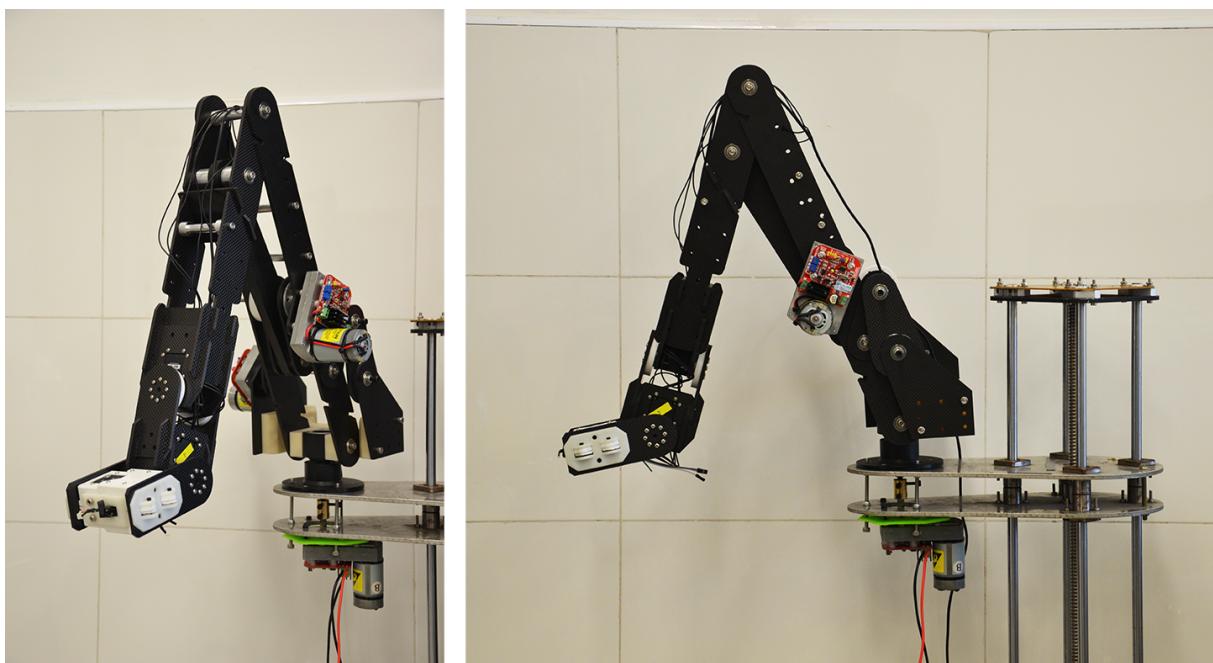


图 32: 第二版机械臂实物图(仍在制作中)

### 4.2 测试结果

图像处理的测试结果见图5,6,7,8,9,10,11,12所示。

立体视觉获得的点云处理结果如图14,15,16,17,18,19所示。



## 5 发展前景与应用价值

我们的机械手能够准确的识别物体并抓取，填补了目前家庭机器人市场中缺乏物体移动功能的空缺。在具有了识别并抓取物体的能力后，机械手能够完成许多基本而有必要的家居任务，比如从柜子上取下一个指定的东西送到手边，或者把桌上杂乱的物体排列整齐，在实际的日常生活中也具有较大的应用价值。

## 6 总结

### 6.1 项目进展

目前，我们的项目已经调试通过了所有模块，也在联合调试中测试了整体的性能。我们的机械臂能够准确的识别物体并抓取，最后将其放在指定的地方，达到了预期效果。现在，我们已经根据第一版机械臂的不足设计制造了第二版机械臂，将在之后的一个月内进行调试。项目进度见表1。

时间	进度
2015.9-2015.10	Kinect 测试 Kinect 物品识别算法开发
2015.11	项目立项，进行可行性分析
2015.12	电机选型、购买与测试
2016.1	初版机械臂设计与加工 机械臂控制算法开发 末端摄像头物品识别算法开发
2016.3	第二版机械臂设计与加工 加入移动平台

表 1：项目进度

## 6.2 展望

我们的系统还有若干方面有改善与提高的空间。我们希望增加机械手的自由度,配合图像算法,使其能够从最合适的方向抓取物体,增加在实际场景中的适应能力;使用如激光雷达来知周围的环境,让我们的机械臂能够更加灵活自主的在房间中移动;语音系统与人脸识别系统,让机械臂能够与人更好的交互。

希望我们的机械臂能走近更多的家庭,帮助人们分担更多的家务,让人们家居生活更加简单轻松。

## 参考文献

- [1] Seungsu Kim, Ashwini Shukla, and Aude Billard. Catching objects in flight. *Robotics, IEEE Transactions on*, 30(5):1049-1065, 2014.
- [2] Carnegie Mellon University. National robotics engineering center. 2016. <http://www.nrec.ri.cmu.edu/projects/arms/>.
- [3] Aitor Aldoma, Zoltan-Csaba Marton, Federico Tombari, Walter Wohlkinger, Christian Potthast, Bernhard Zeisl, Radu Bogdan Rusu, Suat Gedikli, and Markus Vincze. Point cloud library. *IEEE Robotics & Automation Magazine*, 1070(9932/12), 2012.
- [4] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210-227, 2009.
- [5] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105-116. Springer, 1978.