# 文件说明

* 2018.10.23：zsx\_PC=>备份实验室PC端的厦门科技局项目文件
* 总结和整理整个项目，理清楚整个项目的流程，并规划项目的下一步

## 项目总体要达到的目标：

1. 精度的保证：实现三个图像的精准定位，提高精度：

* （1）精度包括三个精度：机械手端精度，定标精度，图像处理之后的物体精度

1. 论文成果转化：把论文中的超分辨率重建的方法引入到实际的系统中
2. 速度问题：在课题要求的速度下，实现运行

***20181025-任务***：

1. 备份当前目录项目的文件（DONE）
2. 简化代码，只显示长方形，正方形和
3. 圆形（DONE）
4. 将三个相机的图像处理都加进去（）
5. 两个相机的位置定标问题，位置坐标解决方案,两套定标程序
6. 是否需要在通讯端增加反馈
7. 速度问题，提速

***20181031代码更新***

* 将所有的模块就精简了（DONE）
* 加入了C相机的定标模块（***未完成***：还没有测试）
* A盘中的长方形，正方形和圆形的区分要与C盘中的长方形，正方形和圆形对应
* ***11月1号需要做的***：
  + 将B相机的程序完成，查找代码，保存图像，处理图像(需要查看编程的flycapture的源代码，看看如何保存图片加载图片)
  + 将C相机的定标程序测试完成
  + Modbus的双工通讯程序要完成

***20181101代码更新***

* 将C相机的图像显示出来了，标定还没进行，今天优化了图像处理，将圆形更好的显示了出来
* ***问题：***在边缘提取之后有很多的边缘，需要选一个，只能是最优的那个，怎么选，是个问题

***20181104代码更新***

* 更新：屏蔽掉了定时器，直接在开始 JAI 相机的时候就直接进行图像的保存和处理
* 总体思路：处理A相机的图像，得到一个坐标，然后直接将值传给机械手(**考虑一个传值的问题，先传圆形（现在想默认先传圆形）还是先传长方形或者正方形（谁面积大就选谁）**)，监听B相机反馈的值（B相机初始值我设为0，如果B相机反馈的值是一个非0，比如1代表是长方形，2代表正方形，3代表圆形），根据返回的值，直接在C相机中找到相应形状的底座坐标
* 具体的解决方案：

1. 在A相机处理程序中，直接将抓取顺序定下来，首先设置Flag == 0,（**public*X和public*Y这两个值是最后我要传给Modbus的值，这里要赋值**）直接传圆形坐标位置给modbus，然后将Flag设置成为非0的值
2. 这里就需要重新写A相机的处理程序，将顺序排出来：**使用面积排序，先看看装矩形的list里面的形式是什么样子的？**
3. ***B相机的程序还是一段空白***
4. C相机要跟随B相机的判定结果来编程

***明天要做的***：

1. 将A相机中的长方形和正方形选出来，利用面积
2. C相机中的长方形和正方形也选出来
3. B相机的程序写入

***20181106代码更新：***

* 优化了代码，将A物料盘中没有物件时的情况考虑了进去，代码健壮性增强
* C物料盘中代码优化还没做，还要考虑长方形，正方形和圆形的情况
* B相机的增加代码

***20181115代码更新：***

* 先完成c盘中没有物件的情况（done）
  1. 区分出了长方形和正方形，通过比较他们的宽的长度确定形状
  2. 通过B相机的识别得到Flag\_B的取值：
  + //TODO:Flag\_B标志位表示B相机是否检测到了物体的形状，
  + //为1则表示检测到了长方形，
  + //2则表示检测到了正方形，
  + //3则表示检测到了圆形
* 加B的代码

***20181116代码更新：***

* 增加了C相机的代码鲁棒性，在没有灯光或者检测不到的情况下的应对策略（done）
* 增加了滤波算法将其他多余的噪声都过滤掉了，使得检测的精度和准确率都有所提升（done）
* C相机的定标（processing）(done)

***20181117代码更新：***

* C相机的定标没问题（done）
* A相机的定标重新设定（done）

***20181118代码更新：***

* 更新了机械手端的代码逻辑
* B相机的代码更新完成（done，检测物件没问题）
* A相机的定标更新也完成(done，解决了相机的中物件消失的情况)
* C相机的定标更新也完成(done，解决了相机的中物件消失的情况)
* 三个相机有时会冲突的Bug问题（修复Bug中）
* 项目要求精度是：0.5MM
  + 机械手的精度是0.001MM（满足要求）
  + A相机和C相机的定标精度都是0.1MM（满足要求）
  + 图像处理后的精度因为是找中心点，所以精度是与各个相机的定标精度相一致（满足要求）
  + B相机的精度还在测量和代码改进中（未完成）

***接下来要完成的任务：***

1. 在B相机的精度问题（待解决）
2. 在B相机的图像处理端，检测到不同形状之后的Flag值的选取情况（通过Flag的取值决定传给机械手的坐标值）
3. 将重建成果加入视觉系统中（正在想办法）
4. 速度问题（此问题相对比较容易）

***Git的使用说明***

***Operation:***

git add .

git commit -m "XXXXXXX"

git push

\*\*\*Quick setup — if you’ve done this kind of thing before\*\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
（1)…or create a new repository on the command line  
 echo "# Project\_xiamen" >> README.md  
git init  
git add README.md  
git commit -m "first commit"  
git remote add origin https://github.com/skywalkerair/Project\_xiamen.git  
git push -u origin master  
（2）…or push an existing repository from the command line  
 git remote add origin https://github.com/skywalkerair/Project\_xiamen.git  
git push -u origin master  
（3）…or import code from another repository  
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.