**目 录**

[6.29 6](#_Toc149742086)

[6.30 6](#_Toc149742087)

[周报（6.29-6.30） 6](#_Toc149742088)

[周报（7.3 -7.7） 7](#_Toc149742089)

[工作安排（7.10 – 7.14） 8](#_Toc149742090)

[7.10 9](#_Toc149742091)

[1. 对涂胶工艺轨道算法进行改进 9](#_Toc149742092)

[2. 细看open3D文档（未看完） 9](#_Toc149742093)

[7.11 10](#_Toc149742094)

[1. 了解工程技术线 10](#_Toc149742095)

[7.12 10](#_Toc149742096)

[7.13 10](#_Toc149742097)

[1 （环境配置）工作&问题记录 10](#_Toc149742098)

[1.1 Windows+vscode+pcl 10](#_Toc149742099)

[1.2 Ubuntu+PCL安装问题 12](#_Toc149742100)

[7.14 13](#_Toc149742101)

[1. 环境配置 13](#_Toc149742102)

[1.1惨败案例 – windows+vscode+PCL 13](#_Toc149742103)

[1.2 Ubuntu安装/远程图形化界面 13](#_Toc149742104)

[2. 论文 - 基于3D机器视觉的工业机器人跟踪涂胶系统 15](#_Toc149742105)

[1.1 \*了解 – 绪论部分相关文献： 15](#_Toc149742106)

[小结 15](#_Toc149742107)

[周报（7.10-7.14） 16](#_Toc149742108)

[工作安排（7.17-7.21） 17](#_Toc149742109)

[7.17 17](#_Toc149742110)

[1.通过sdk方式获取海康相机数据 17](#_Toc149742111)

[2. 环境配置（ubuntu+vscode windows+vs）PCL 17](#_Toc149742112)

[2.1 cmake安装 17](#_Toc149742113)

[2.2 vscode部署pcl（ubuntu-cmake） 17](#_Toc149742114)

[2.3 windows+vs+PCL 18](#_Toc149742115)

[7.18 18](#_Toc149742116)

[1. 论文 - 基于线激光的水轮机机器人测...标定与焊点加工区域特征提取 18](#_Toc149742117)

[1.1 \*手眼标定 --- 相关文章 19](#_Toc149742118)

[1.2 \*点云处理 19](#_Toc149742119)

[7.19 19](#_Toc149742120)

[1. \*点云处理 – 昨天论文 20](#_Toc149742121)

[2. C++ & PCL学习 20](#_Toc149742122)

[2.1 区域生长分割 20](#_Toc149742123)

[2.2 提取聚类分类 20](#_Toc149742124)

[2.3 长边提取示例 21](#_Toc149742125)

[3.windows+vscode+pcl+cmake 问题剖析 21](#_Toc149742126)

[4. 海康相机 – 调研+实践 22](#_Toc149742127)

[4.1使用SDK拿去海康相机数据 22](#_Toc149742128)

[4.2 拍出钢板点云图，使用VM进行处理，构建基本思路 22](#_Toc149742129)

[小结： 22](#_Toc149742130)

[7.20 23](#_Toc149742131)

[1. 钢板项目 23](#_Toc149742132)

[1.1 已解决 - 问题1：项目场景 23](#_Toc149742133)

[1.2 已解决 - 问题2：获取点云之后的工作如何进行 23](#_Toc149742134)

[2. 论文 - 基于线激光的水轮机机器人测...标定与焊点加工区域特征提取 24](#_Toc149742135)

[2.1 (章节3.2.3)线激光点云去重算法分析与实现 24](#_Toc149742136)

[2.2 （章节3.3）线激光点云焊点特征分割算法研究与改进 25](#_Toc149742137)

[小结： 25](#_Toc149742138)

[3. 改进 - 手机边框点云提取 26](#_Toc149742139)

[4. 已解决 - windows+vscode+PCL 26](#_Toc149742140)

[5. 关于BUG - 小结 27](#_Toc149742141)

[7.21 28](#_Toc149742142)

[1. Halcon 28](#_Toc149742143)

[1.1 安装 – 并在vs调用halcon的包 28](#_Toc149742144)

[2. pcl学习路线及资料 28](#_Toc149742145)

[3. 海康相机 – 数据信息 28](#_Toc149742146)

[3.1 点云数据 – 深\亮度图 28](#_Toc149742147)

[周报（7.17-7.21） 29](#_Toc149742148)

[工作安排（7.24-7.28） 29](#_Toc149742149)

[7.24 29](#_Toc149742150)

[1. 可行性报告 – 论文 29](#_Toc149742151)

[1.1 工业机器人涂胶路径规划与仿真研究 29](#_Toc149742152)

[1.2 基于视觉的机器人自动化涂胶质量检测技术研究 30](#_Toc149742153)

[7.30 30](#_Toc149742154)

[1 使用PCL进行点云提取 30](#_Toc149742155)

[2. Open3D安装 30](#_Toc149742156)

[3. 手机胶道改进 31](#_Toc149742157)

[7.31 34](#_Toc149742158)

[1.Vscode+Docker+PCL 34](#_Toc149742159)

[8.1 34](#_Toc149742160)

[1.区域生长 34](#_Toc149742161)

[2.计算相机参数excel表格 34](#_Toc149742162)

[8.2|8.3 35](#_Toc149742163)

[1.充分必要条件 35](#_Toc149742164)

[2. 论文 - 基于线激光的水轮机机器人测...标定与焊点加工区域特征提取 35](#_Toc149742165)

[2.1手眼标定 35](#_Toc149742166)

[8.14 35](#_Toc149742167)

[1.论文-基于标准圆柱的线激光轮廓扫描机器人手眼标定方法 35](#_Toc149742168)

[1.1 知识点 35](#_Toc149742169)

[1.2 论文笔记 37](#_Toc149742170)

[1.3 论文待解决问题 38](#_Toc149742171)

[2. 论文 - 线激光器的手眼标定方法 38](#_Toc149742172)

[2.1 四元数 – 含未解决问题 38](#_Toc149742173)

[3. 手眼标定论文小结 38](#_Toc149742174)

[8.18 38](#_Toc149742175)

[1.wsl & Docker 38](#_Toc149742176)

[工作安排(8.21-8.25) 38](#_Toc149742177)

[8.21 39](#_Toc149742178)

[1.最小二乘法 39](#_Toc149742179)

[8.22 39](#_Toc149742180)

[1.手眼标定相关知识 39](#_Toc149742181)

[1.1 欧拉角-四元数-旋转矩阵 39](#_Toc149742182)

[8.23 39](#_Toc149742183)

[1. 点云 – 网络 39](#_Toc149742184)

[8.28 39](#_Toc149742185)

[1. 手眼标定相关博客 39](#_Toc149742186)

[8.30 39](#_Toc149742187)

[1. vs问题 – 无法Debug 39](#_Toc149742188)

[2. vs问题 – 使用pcl智能指针报错 – 错误代码-1073740940 39](#_Toc149742189)

[8.31 40](#_Toc149742190)

[1. PointNet – 包含论文解读 40](#_Toc149742191)

[9.1 40](#_Toc149742192)

[1.关于点云提取 40](#_Toc149742193)

[2. 提取成功 40](#_Toc149742194)

[3. 点云提取截图 42](#_Toc149742195)

[9.15 42](#_Toc149742196)

[1. 手眼标定相关博客 42](#_Toc149742197)

[10.8 42](#_Toc149742198)

[1.失败 - 搭建socks5代理服务器 42](#_Toc149742199)

[2. 突然无法使用conda命令 42](#_Toc149742200)

[10.10 42](#_Toc149742201)

[1.关于访问某些网页很慢的问题 42](#_Toc149742202)

[10.18 43](#_Toc149742203)

[求解手眼矩阵(TSAI & PSO) 43](#_Toc149742204)

[胶道点云中心线提取 43](#_Toc149742205)

[FRP实现内网穿透 43](#_Toc149742206)

[使用TSAI和PSO分别求解手眼矩阵 43](#_Toc149742207)

[10.27 44](#_Toc149742208)

[Wsl配置深度学习 44](#_Toc149742209)

[10.31 44](#_Toc149742210)

[1. vscode配置多个分支(实现多存储库) 44](#_Toc149742211)

[2. 前端预览DOCX文件 44](#_Toc149742212)

# 6.29

* VM文档全局变量模块

全局变量在模块结果中订阅参数和在全局变量模块订阅目标输出

* 可以看一下别人实现VM文档案例时的笔记[visionmaster- CSDN搜索](https://so.csdn.net/so/search?q=visionmaster&t=blog&u=liyuanbhu)
* 3DMVS还剩数据块控制和传输层控制没有看，不知道有用没
* 学会了如果进行拼图展现深度图、3D点云图

使用触发控制，在没有外接编码器的情况下可以使用帧触发的软触发来实现出图

* 通读了一遍VM文档，没读完，实现了部分案例

# 6.30

* 3DMVS和VM的数据传输关系，VM上好像没办法操控相机进行取流，那么是需要使用3DMVS取到自己需要的图像之后，再打开VM进行图像处理吗？
* 应该是可以通过相机参数模块对相机进行调参然后取流
* 

尝试了一下全局触发以及VM的通讯功能

* 对于VM流程图的一些部件和其中的参数还是比较混沌

# 周报（6.29-6.30）

对相机进行实操

组装相机过程遇到了电源适配器电压不足无法正常启动的情况

练习了3DMVS的取流操作以及触发控制

在VM平台使用2D图做了一些参考案例

周报（7.3 -7.7）

通读了open3D的文档，并了解了一些其他的关于点云处理的python第三方库

通过手机涂胶工艺轨迹提取，学习到了点云的基本操作，包括点云滤波预处理，点云分割和聚类。

手机涂胶工艺轨迹提取目前效果不太理想，第一就是泛化性太弱，针对性太强，都是针对已有的点云数据进行滤波分割，没有通用性。

就目前手里的数据而言已有成果就是短边提取较为成功，长边的点云较难提取。

|  |  |
| --- | --- |
|  |  |
| 长边偶尔会提取成功 | 提取长边时，在进行分割前后进行统计->半径滤波---------------还是随缘 |

# 工作安排（7.10 – 7.14）

\*1. 对涂胶工艺轨道算法进行优化 – 尝试新的方法

虽然没有尝试新的方法，但是基本实现了功能

\*2. 尝试进行点云拼接

\*3. 了解手眼标定算法

\*4. 了解机器人轨迹生成为主，重点关注轨迹规划及图形化交互

\*5. 进行初步的可行性分析

6. 了解PCL和一些其他处理点云的第三方库

7. 了解目前技术的研究现状

8. 出一份初步的可行性报告

# 7.10

日工作计划：

## 1. 对涂胶工艺轨道算法进行改进

第一版：平面分割->聚类分割->滤波

第二版：滤波->平面分割->聚类分割->滤波->(聚类/平面分割)

最终版：聚类分割->平面分割->滤波->聚类分割

最终版缺陷：对于点云残缺较多的点云图无法提取工艺轨道

预期解决方案：首先进行点云缺陷填补

\*\*\*\*\*\*\*\*最终版方案对于左边长边来说不行，左长边(long2)需要在第二次滤波选择第一大簇(解决方案目前没有想好)

|  |  |
| --- | --- |
|  |  |

## **2. 细看open3D文档（未看完）**

在cvtutorials.com细看了open3D文档，没看完

# 7.11

1. 了解工程技术线

找一些论文，了解当前技术发展（可以再改进一下轨道提取代码）

# 7.12

Ubuntu官方的虚拟机multi pass有bug，后来选择了virtualbox

# 7.13

安排：学习ubuntu，PCL。画出项目流程图并使用VM模拟项目流程。博士论文读完。了解其他工具站。查看群宾专利。

## 1 （环境配置）工作&问题记录

### 1.1 Windows+vscode+pcl

|  |
| --- |
| 两个都需要安装，pdb那个文件解压缩到PCL\bin 里  C:\Users\austin.zhang\DEV\PCL\_1\_13\_0\3rdParty\OpenNI2  这个文件可能会安装到C:\Program Files |
| 6个环境变量 |

Windows下使用vscode安装pcl很不友好，目前存在问题：识别不了头文件，再includePath中添加路径也没有用 || 感觉includePath知识为了头文件不全

|  |
| --- |
| 路径只是不对，路径指示在当前路径下    使用\*\*递归，至少头文件路径没问题 -- 可能需要重启vscode |
| 补充：  Includepath只是帮助插件找到文件，而报错找不到文件是因为编译找不到，所以要为编译器指定文件路径。（！！！！必要的时候关闭coderrunner， 他的报错信息不完整）      （coderunner）  这些库是相互依赖的，关闭coderunner后可以看到清晰的报错，一个一个添加依赖    这是新的问题，提示平台不支持aligned\_malloc和aligned\_free  之前mingw下载成win32版本的了，更换mingw版本后pcl的头文件总是报错，放弃 |

* 关于virtualBox虚拟机启动异常和无法保存设置 ————— 使用管理员权限启动

### 1.2 Ubuntu+PCL安装问题

|  |
| --- |
| * sudo apt-get update * sudo apt-get install git build-essential linux-libc-dev * sudo apt-get install cmake cmake-gui * sudo apt-get install libusb-1.0-0-dev libusb-dev libudev-dev * sudo apt-get install mpi-default-dev openmpi-bin openmpi-common * sudo apt-get install libflann1.9 libflann-dev * sudo apt-get install libeigen3-dev * sudo apt-get install libboost-all-dev * sudo apt-get install libqhull\* libgtest-dev * sudo apt-get install freeglut3-dev pkg-config * sudo apt-get install libxmu-dev libxi-dev * sudo apt-get install mono-complete * sudo apt-get install libopenni-dev * sudo apt-get install libopenni2-dev   批量安装了这些依赖，不知道有用没  Vim xx.sh -> sh xx.sh（就可以批量执行命令） |
| sudo apt install libpcl-dev  看了另一个教程，一个命令就可以了，后续具体操作看下面链接  [(145条消息) Ubuntu20.04 Ubuntu18.04安装pcl点云库\_长沙有肥鱼的博客-CSDN博客](https://blog.csdn.net/weixin_53660567/article/details/119871890) |

Ubuntu可在终端运行PCL，windows未成功

# 7.14

## 1. 环境配置

### 1.1惨败案例 – windows+vscode+PCL

\*针对windows环境使用vscode开发pcl的环境配置问题，应该从cmake的配置和编译器在编译过程中寻找头文件路径下手

\*先尝试使用cmake-gui编译，看是否能通过

（windows + vscode + pcl）太麻烦，windows使用vs， ubuntu使用vscode

### 1.2 Ubuntu安装/远程图形化界面

|  |
| --- |
| Ubuntu安装图形化界面及远程连接（VNC登录）  # 进入root  sudo -s  # update一下  apt-get update  # 安装x-winodws  sudo apt-get install x-window-system-core  # 安装登陆管理器  sudo apt-get install gdm3  # 安装ubuntu桌面  sudo apt-get install ubuntu-desktop  # 安装gnome套件  sudo apt-get install gnome-panel gnome-settings-daemon metacity nautilus gnome-termi  4.然后输入  reboot重启  ————————————————  原文链接：https://blog.csdn.net/wuyihao123/article/details/127479795） |
| # vim ~/.vnc/xstartup  这个配置可以出来文件夹，但是只能出来文件夹  export XKL\_XMODMAP\_DISABLE=1  unset SESSION\_MANAGER  unset DBUS\_SESSION\_BUS\_ADDRESS  #!/bin/sh  # Uncomment the following two lines for normal desktop:  # unset SESSION\_MANAGER  # exec /etc/X11/xinit/xinitrc  [ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup  [ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources  xsetroot -solid grey  vncconfig -iconic &  x-terminal-emulator -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &  x-window-manager &  gnome-panel &  gnmoe-settings-daemon &  metacity &  nautilus &  gnome-terminal & |
| VNC远程图形化界面好像真的只有文件夹而不是整个桌面，遂放弃。  解决方案，windows(mstsc) + ubuntu(xrdp) 需要设置root用户和密码，刚开始连接很卡顿，应该在windows界面性能里面转换一下网络。  此外，根据下面的文章对xrdp进行了优化。  [(146条消息) Xrdp 体验优化 减少/解决画面卡顿\_xrdp卡顿\_wuweijie@apache.org的博客-CSDN博客](https://blog.csdn.net/wu_weijie/article/details/116158271)  刚开始查的是只能用root用户登录，后来莫名其妙的其他用户也可以登录了 |

## 2. 论文 - 基于3D机器视觉的工业机器人跟踪涂胶系统

### 1.1 \*了解 – 绪论部分相关文献：

1.本文依托于AI 工业视觉公司“北京阿丘科技有限公司”的生产研发项目

2.哥伦比亚大学P．ALLEN 跟踪定位于抓取系统 该系统为Puma560 机器人增添加了捕捉移动目标的实时视觉

3. 日本FAUNC 机器人公司研发的“高速双臂工业机器人分拣系统”

4. Murakami．S 等学者研究了基于模糊逻辑控制器的焊缝跟踪控制系统

Murakami S, Takemoto F, Fujimura H, et al. Weld-line tracking control of arc welding robot using fuzzy logic controller[J]. Fuzzy Sets and Systems, 1989, 32(2): 221-237.

5. Jae Seon Kim[15]和W. P. Gu[16]分别提出了基于线结构光三维测量系统的工业机器人焊接轨迹跟踪的方法

* 采取D-H 法进行涂胶机器人的运动学建模

|  |
| --- |
|  |
|  |

* 这篇论文具体技术细节帮助不大，但是里面的参考文献和提到的方法可以一看

## 小结

不要将目光局限于市面上的涂胶技术，类似的焊接等技术也要多看

在撰写论文的过程中，重点肯定是项目的创新点和技术难点，但是所使用的现有技术也可以对其进行剖析，例如本项目的3D扫描相机，以及以后会改进的一些算法，在撰写论文的时候可以从其优缺点、性能等方面进行展开。

# 周报（7.10-7.14）

\*1. 对涂胶工艺轨道算法进行优化 – 尝试新的方法

虽然没有尝试新的方法，但是基本实现了功能

\*2. 尝试进行点云拼接

\*3. 了解手眼标定算法

\*4. 了解机器人轨迹生成为主，重点关注轨迹规划及图形化交互

\*5. 进行初步的可行性分析

6. 了解PCL和一些其他处理点云的第三方库

7. 了解目前技术的研究现状

8. 出一份初步的可行性报告

上周任务基本没有完成，主要是被很多bug和环境配置缠住，在使用ubuntu的multipass时浪费太多时间，配置windows+vscode+pcl也浪费了很多时间，在解决一个新问题的时候设定一个目标时间，如果没有解决就需要换一个解决办法，或者暂停一下。

总结：找了一些开源虚拟机，在上面部署了基于ubuntu的pcl开发环境，了解了一下pcl处理点云的操作。针对市面上一些可以进行3d点云处理任务的软件进行了简单调研，类似于VM之类的。找到了一些比较贴合项目的论文，目前正在研读和归纳总结。

# 工作安排（7.17-7.21）

1. 配置windows下的pcl环境，熟悉一下c++和pcl操作，就拿手机涂胶工艺轨道进行练手
2. 尽快进行可行性分析，思路：完整项目的论文、群宾等公司的专利、相关技术点的论文、市面上相关的软件(VM， Halcon)
3. 三人进行讨论，争取拿出初步的可行性报告

# 7.17

工作安排：对VM和Halcon进行调研(使用VM做一下手机涂胶轨道提取 – 这玩意只能导入.raw格式的本地图像，后续考虑使用Halcon吧)，根据相关论文画出初步的对于项目理解的流程图。在Ubuntu环境下尝试学习PCL和c++。可以考虑配置windows环境下的PCL环境。学会如何获取海康相机数据。

## 1.通过sdk方式获取海康相机数据

|  |
| --- |
| C:\Users\austin.zhang\DEV\3DMVS\Development\Mv3dLpSDK  感觉比较复杂，回头可以跑一下官方给的样例  [(146条消息) C#快速调用海康威视工业相机的SDK拍照获取图片\_zls365365的博客-CSDN博客](https://blog.csdn.net/zls365365/article/details/121918147)  可以参考这个例子 |

## 2. 环境配置（ubuntu+vscode windows+vs）PCL

### 2.1 cmake安装

|  |
| --- |
| [ubuntu 20.04安装(升级)cmake - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/519732843)  [(146条消息) ubuntu安装cmake\_yuanzhoulvpi的博客-CSDN博客](https://blog.csdn.net/yuanzhoulvpi/article/details/122938078?ydreferer=aHR0cHM6Ly9jbi5iaW5nLmNvbS8%3D)  **下回要安装新的东西时，记得查看是否有旧版本的。这次安装是cmake3.25，又是可以make成功，有时候显示当前运行版本不是cmake3.25，运行失败** |

### 2.2 vscode部署pcl（ubuntu-cmake）

|  |
| --- |
| **CmakeLists.txt**  cmake\_minimum\_required(VERSION 3.25 FATAL\_ERROR)  project(pcl\_test2)  find\_package(PCL 1.10 REQUIRED)  include\_directories(${PCL\_INCLUDE\_DIRS})  link\_directories(${PCL\_LIBRARY\_DIRS})  add\_definitions(${PCL\_DEFINITIONS})  add\_executable(pcl\_test2 pcl\_test2.cpp)  target\_link\_libraries(pcl\_test2 ${PCL\_LIBRARIES}) |
| 配置了Cmake就不用配置其他文件了（包括includepath）更详细操作如下：  [(146条消息) ubuntu 上vscode使用cmake编译运行c++程序\_ubuntu vscode c++\_SCH0的博客-CSDN博客](https://blog.csdn.net/qq_41246375/article/details/119546955#%E5%89%8D%E6%8F%90) |

### 2.3 windows+vs+PCL

|  |
| --- |
| [Windows11+VS2022+PCL1.13.0 安装配置记录\_WoooChi的博客-CSDN博客](https://blog.csdn.net/m0_50910915/article/details/128651063)  属性文件可以重复利用，不用每次都设置。  项目–属性–调试–环境–编辑—添加环境如下(每次都要设置)  PATH=$(PCL\_ROOT)\bin;$(PCL\_ROOT)\3rdParty\FLANN\bin;$(PCL\_ROOT)\3rdParty\VTK\bin;$(PCL\_ROOT)\3rdParty\Qhull\bin;$(PCL\_ROOT)\3rdParty\OpenNI2\Tools;$(PATH) |

问题1. 运行简单生成的点云文件没有问题，读取pcd文件会报错

|  |
| --- |
| Io.h必须在cloud\_viewer.h下面，不然io.h就会报错 |

# 7.18

工作安排：1.研读基于线激光的水轮机机器人测...标定与焊点加工区域特征提取\_吉鹏晖，并结合之前关于涂胶的论文，整理出整个项目的大概思路。2.调研VM、Halcon和市面上其他商用的视觉算法。3.学习PCL，整理PCL和open3D的差异4.调研海康相机

（将这篇文档转换为markdown，下载git，以后直接上传云端）

## 1. 论文 - 基于线激光的水轮机机器人测...标定与焊点加工区域特征提取

文中有很多参考文献，和国内外相关技术的研究路线

### 1.1 \*手眼标定 --- 相关文章

|  |
| --- |
| 1. 感觉文章很好，但是没有看懂  [(147条消息) 工业机器人工具坐标系（TCF）标定的六点法原理\_工业机器人tcp六点法\_-贝塔-的博客-CSDN博客](https://blog.csdn.net/qq_41658212/article/details/105686309)  2.C:\Users\austin.zhang\Documents\Paper\reading  一种结合 TCP 标定的深度相机手眼标定方法  3.超级详细！！  [(147条消息) 机器人手眼标定Ax=xB（eye to hand和eye in hand）及平面九点法标定\_yaked19的博客-CSDN博客](https://blog.csdn.net/yaked/article/details/77161160)  4.博主的文章比较全面  [(147条消息) 机器人手眼标定原理介绍（含详细推导过程）使用Tsai-Lenz算法\_鱼香ROS的博客-CSDN博客](https://blog.csdn.net/qq_27865227/article/details/114011388) |
| 1. AX=XB问题的计算 --- 精度 |

### 1.2 \*点云处理

|  |
| --- |
| 点云分割算法主要有以下四大类：基于点云聚类的、基于区域生长的、基于边缘识别的、基于标准模型的。  基于点云聚类的算法中欧式聚类使用较为广泛，其适合在已经分割划分之后对剩下的部分点云进行聚类以便后续的进一步分割。基于边缘识别的方法则适用于在分割刚开始时先利用边缘信息进行预分割。基于标准模型的方法有 Hough 变换和RANSAC（随机样本一致性），其中 Hough 变换适合提取直线、圆等特征， RANSAC则多用于平面、圆柱面、球面等。基于区域生长的分割则关注点云中点与点之间的特征信息，通过这样的特征信息将点与点进行合并或者分开。 |

# 7.19

工作安排：1.研读基于线激光的水轮机机器人测...标定与焊点加工区域特征提取\_吉鹏晖，并结合之前关于涂胶的论文，整理出整个项目的大概思路。2.调研VM、Halcon和市面上其他商用的视觉算法。3.学习PCL，整理PCL和open3D的差异4.调研海康相机

（将这篇文档转换为markdown，下载git，以后直接上传云端）

## 1. \*点云处理 – 昨天论文

|  |
| --- |
| 从左至右依次为欧式聚类、区域生长、边缘识别、 RANSAC 算法的效果在点云分割中除了上述分割方法外还有对点云的语义分割（PCSS）， PCSS 相比于 PCS 会在分割后为每个点生成语义信息。 PCSS 的常规的方法为有监督机器学习，目前较为前沿的还有点云的深度学习方法 |

## 2. C++ & PCL学习

### 2.1 区域生长分割

[(149条消息) pcl小知识（四）——区域生长分割原理(region growing segmentation)\_pcl分割\_刘坤的博客的博客-CSDN博客](https://blog.csdn.net/liukunrs/article/details/80482788)

|  |
| --- |
| 看着效果还可以，应该可以实现提取优化 |

### 2.2 提取聚类分类

[(149条消息) pcl点云聚类后的点云索引提取与输出pcd聚类结果\_菜是菜人是真帅的博客-CSDN博客](https://blog.csdn.net/uranus1992/article/details/83344952)

|  |
| --- |
| 提取最值 |

### 2.3 长边提取示例

|  |
| --- |
| open3d聚类long\_utility.pcd -> pcl区域生长long\_grow0.pcd |

## 3.windows+vscode+pcl+cmake 问题剖析

|  |
| --- |
|  |
| 莫名其妙，环境变量和CmakeLists.txt都设置了PCL\_ROOT，结果还是错的 |

## 海康相机 – 调研+实践

### 4.1使用SDK拿去海康相机数据

|  |
| --- |
| 海康有给简单的示例，但是功能不多 |

### 4.2 拍出钢板点云图，使用VM进行处理，构建基本思路

|  |
| --- |
| ？钢板横截面拿相机一扫就出来了，之后滤波一下噪声即可。  对项目要做什么还是不清楚 |

## 小结：

又浪费时间调试了一下widows+vscode+pcl，找出了问题在哪，但不知道为什么有问题。论文看了一点，看到论文采用的点云分割方法后就去动手实践，没有完全做出来，但是感觉效果应该比open3D好些。重新连接了海康相机，拍出了钢板的点云图，后续使用VM进行一些处理，尝试了一下海康自带的SDK实例，感觉功能有点少，需要自己写。明天的话最起码把论文基本过一遍，然后找几篇文献，把流程图和可行性报告的首稿做出来。

# 7.20

工作安排：1.搞清钢板点云处理的任务需求2.使用PCL和open3D对点云提取进行优化3.论文看完并整理思路，使用chatgpt整理出可行性报告初版。

待解决问题：1.钢板的详细场景需求2.获取点云之后，机械臂如何根据提取的点云进行后续工作

\*每天安排的任务量力而行，总结的时候要有目的的总结，应该是通过什么学会了什么，还有什么疑问，后续待解决的问题，然后可以说一些注意事项。针对一样意义不大的工作不应该浪费太多时间，比如windows+pcl+vscode，半个小时如果解决不了就可以先放下等有时间再解决，搞清楚干一件事的目的，配置这个环境是为了学习pcl而不是为了配置环境而配置。

## 1. 钢板项目

### 1.1 已解决 - 问题1：项目场景

钢板是竖着放在钢筋上面，然后在钢板上面覆盖一层，并将其与钢板进行焊接，那么在工作时下面与钢筋是怎样连接的，已经焊接好的吗，在后续焊接中还需要考虑下面的情况吗？

|  |
| --- |
| 首先，这个工作的任务场景是针对钢板上边缘的焊接，那么针对点云提取就好  关键点在于能不能使用VM将一套工作直接做出来 |

### 1.2 已解决 - 问题2：获取点云之后的工作如何进行

获取点云轨迹之后如何根据点云轨迹进行工作，不论是涂胶还是焊接。

|  |
| --- |
| 获取点云之后需要根据点云进行关键点提取，然后传给机器人进行机器人轨迹生成 |

## 2. 论文 - 基于线激光的水轮机机器人测...标定与焊点加工区域特征提取

### 2.1 (章节3.2.3)线激光点云去重算法分析与实现

#### 2.1.1 FPS算法

|  |
| --- |
| [Farthest Point Sampling (FPS)算法核心思想解析 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/114522377)  使用了动态规划的思想。  上图解释的很明白，L数组是A中点到点P的距离（点P是每次新加入的点）  算法的原用和延伸：  点云去重的目的是要去除距离极为接近的点，现有算法中Farthest Point Sampling(FPS)可以实现这种目的。FPS 算法流程如图3-7 所示，虽然FPS 的设计初衷是为了进行降采样，但其“每次从集合 A 中选一个点，使得其到集合 B 里面点的距离最大”的思想可直接用于点云去重。  弊端：  然而FPS 算法受采样点预设值k 的影响很大，k 值设置的不好容易导致点云去重不完全或者点云出现缺失。 |

#### 2.1.2 没看懂 - 使用半径邻域搜索实现去重

|  |
| --- |
|  |

### 2.2 （章节3.3）线激光点云焊点特征分割算法研究与改进

#### 2.2.1 区域生长分割

**该算法的目的是合并在平滑约束条件下足够接近的点。因此，该算法的输出数据结构是由聚类组成的数组，其中每个聚类都是被认为是同一光滑表面的一部分的点的集合。该算法的工作原理（光滑度的计算）是基于两点法线之间的角度比较。**

这句话很重要，之前在对手机边框使用聚类进行提取的时候，总是想着把边框直接给聚出来，但是有时候可能反着来更好，把特征明显的区域聚出来，直接扔掉也可以！例如区域生长是找平滑曲面，工艺轨迹附近的平面都比较光滑，且特征明显，可以针对他们进行提取然后丢弃。

|  |
| --- |
| 搜索到邻域后，这些点先过法线夹角阈值，通过的保留到聚类数据，然后再从这些通过法线夹角阈值的点中，检查是否通过曲率阈值，通过的加入种子点序列。  [(149条消息) 【C++】pcl中的Region Growing（区域生长）算法\_pcl::regiongrowing\_Zhang\_Chen\_的博客-CSDN博客](https://blog.csdn.net/Zhang_Chen_/article/details/101228569) |

### 小结：

文章看完，感觉对于整个项目的认知没有变得更清晰，各部分功能实现还是有些割裂。

在本文涉及的实际场景中，由于线激光测量设备的实际扫描范围有限，当需要获取一个大范围区域的表面点云时，应将多次扫描的结果汇总。在多次扫描时必将出现部分区域进行重复扫描的情况，这就导致最后获得的点云有重复点出现。

## 3. 改进 - 手机边框点云提取

改变了策略，2.2.1部分有相关思想，目前使用open3d进行滤波处理效果不错，但是分割提取部分应该还需要使用PCL来做。

Open3d滤波思想，对原始点云进行聚类 -> 双重滤波 -> 聚类 -> 双重滤波，好处就是减少平面切割带来的不确定性，提高了算法的鲁棒性，同时使得代替去特征区域更加明显。（就是不知道为什么双重滤波部分很慢）

## 已解决 - windows+vscode+PCL

#### 4.1 彻底解决方案

更改vscode中kit位vs Release即可！！！！！！！

#### 4.2 未彻底解决 - MinGW解决方案

|  |
| --- |
|  |
|  |
|  |

修改第二条分支可以找到正确路径，但还是报错。报错信息如下：

|  |
| --- |
|  |

这里的win32指的不是系统，而是编译器版本 这里是错误的，使用64位的kit，依然走的是第一条分支。

## 关于BUG - 小结

不论是配置环境还是平时遇到BUG，需要仔细查看日志以及报错信息去排查错误，不要急，慢慢排查，不要一直在一条死胡同走！

# 7.21

工作安排：1.找相关案例和商用软件写可行性报告2.使用VM对钢板点云进行处理并研究海康相机性能3.对点云提取进行优化 – 顺手把周报写了

## 1. Halcon

### 1.1 安装 – 并在vs调用halcon的包

|  |
| --- |
| [(149条消息) halcon23.05下载安装，并在qt creator和vs2022使用halcon的包\_稷滼的博客-CSDN博客](https://blog.csdn.net/jifanyyds/article/details/130964282) |

## 2. pcl学习路线及资料

|  |
| --- |
| [PCL(Point Cloud Library)学习指南&资料推荐（2023版） - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/268524083)  Github已star |
| [01-点云及其可视化 - 黑马机器人 | PCL-3D点云 (czxy.com)](https://robot.czxy.com/docs/pcl/chapter01/intro/) |
| [pcl- CSDN搜索](https://so.csdn.net/so/search?q=pcl&t=blog&u=qq_43232556) |

## 3. 海康相机 – 数据信息

### 3.1 点云数据 – 深\亮度图

|  |
| --- |
|  |
|  |
|  |

# 周报（7.17-7.21）

1.了解了如何通过sdk直接拿取海康相机数据 2.读完了水轮机的论文，学习了关于整个项目的构建思路3.了解PCL大概的技术功能，并开始细致的学习PCL和C++语法

\*读论文要有侧重点，工作时间应该把精力集中在重要的地方

# 工作安排（7.24-7.28）

1.可行性报告 -> 市场：市面上类似的项目 -> 技术：探索商用视觉算子 -> 技术：本项目要使用的技术。

2.钢板项目，使用VM在嘈杂环境下也能把钢板提出来

3.详细学习C++和PCL

4.看论文

# 7.24

1.针对上周调研的内容和以前看的论文写出可行性报告初版

2.使用VM/Halcon对钢板点云进行提取

3.学习C++和PCL

## 1. 可行性报告 – 论文

### 1.1 工业机器人涂胶路径规划与仿真研究

|  |
| --- |
| 工业机器人技术在智能制造业中的应用越来越广泛，在涂胶领域的应用，可以有效解决人工涂胶效率低、涂胶质量不高、涂胶过程有害气体对人体伤害等问题。通过与传统的工业机器人现场示教点编程生成轨迹路径的方法相比较，采用虚拟仿真软件中“自动路径”功能优化后生成的路径轨迹与几何体曲面轮廓贴合度更好，涂胶厚度更加均匀，涂胶质量和效率更高。  胶接作为一种重要的连接方式，广泛应用于产品的制造过程中。目前，国内仍然有大部分企业采用人工涂胶的方法，手工涂胶的质量受个人熟练程度限制，容易出现涂胶不均匀、不连续和浪费等现象，并且大部分胶易挥发出有毒气体，对人体产生伤害。采用现场示教点位的编程方法可以很好地完成单一平直的涂胶路径轨迹规划，但对于具有复杂曲线的涂胶路径此种方法难以满足要求，会出现示教点位过多、编程工作量大、生产效率低下、涂胶质量差的情况。 |

### 1.2 基于视觉的机器人自动化涂胶质量检测技术研究

|  |
| --- |
| 摘要及绪论部分有很多可以借鉴 |
|  |
|  |

# 7.30

## 1 使用PCL进行点云提取

|  |
| --- |
| <https://blog.csdn.net/luolaihua2018/article/details/120184539>  感觉可以实现，但是BUG太多  https://blog.csdn.net/qq\_25105061/article/details/119614919 |

## Open3D安装

|  |
| --- |
| conda install numpy #安装matplotlib  conda install matplotlib #安装matplotlib  conda install -c open3d-admin open3d #安装Open3D |

## 手机胶道改进

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  | 如上 |

# 7.31

## 1.Vscode+Docker+PCL

|  |
| --- |
| [安装Docker和配置VScode连接 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/368330656) |

# 8.1

## 1.区域生长

|  |
| --- |
| [(149条消息) pcl 区域生长算法（一）\_长沙有肥鱼的博客-CSDN博客](https://blog.csdn.net/weixin_53660567/article/details/120674502)  [(149条消息) pcl小知识（四）——区域生长分割原理(region growing segmentation)\_pcl分割\_刘坤的博客的博客-CSDN博客](https://blog.csdn.net/liukunrs/article/details/80482788) |

## 2.计算相机参数excel表格

|  |
| --- |
| =VLOOKUP($C$3,$P:$R,2,0) 根据C3的数据获取P-R的第二列，C3数据必须在P-R的第一列（0表示精准查找，1表示模糊查找） |
| 将excel表格保存为xlsm即可保存其中的宏或VB |

# 8.2|8.3

## 1.充分必要条件

|  |
| --- |
|  |

## 2. 论文 - 基于线激光的水轮机机器人测...标定与焊点加工区域特征提取

### 2.1手眼标定

|  |
| --- |
| [#手眼标定 (qq.com)](https://mp.weixin.qq.com/mp/appmsgalbum?__biz=MzkzMzI2MTU2Nw==&action=getalbum&album_id=1938760984928550918&scene=173&from_msgid=2247483901&from_itemidx=1&count=3&nolastread=1#wechat_redirect) |
|  |

# 8.14

## 1.论文-基于标准圆柱的线激光轮廓扫描机器人手眼标定方法

### 1.1 知识点

#### 1.1.1 罚函数

|  |
| --- |
| 罚函数的基本思想是，借助罚函数把约束问题转化为无约束问题，进而用无约束最优化方法求解。  [罚函数法 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/397705593) |

#### 1.1.2 齐次变换矩阵

|  |
| --- |
| 齐次变换矩阵，R是旋转矩阵（正交矩阵），p是平移向量， [0 0 0 1]是为了方便运算  Ps: |
|  |

#### 1.1.3 PSO粒子群优化算法

|  |
| --- |
| [粒子群优化算法(Particle Swarm Optimization, PSO)的详细解读 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/346355572) |

#### 1.1.4 ransac拟合椭圆

|  |
| --- |
| import numpy as np  from sklearn.linear\_model import LinearRegression, RANSACRegressor  from sklearn.metrics import mean\_squared\_error  # 生成随机点云  n\_samples = 100  x = np.random.uniform(-10, 10, n\_samples)  y = np.random.uniform(-5, 5, n\_samples)  noise = np.random.normal(0, 1, n\_samples)  x += noise  y += noise  # 将点云转换为二维数组  data = np.column\_stack((x, y))  # 定义椭圆模型  def ellipse\_model(x, a, b, h, k):      return ((x[:,0]-h)/a)\*\*2 + ((x[:,1]-k)/b)\*\*2  # 定义RANSAC回归器  ransac = RANSACRegressor(base\_estimator=LinearRegression(),                           min\_samples=10, residual\_threshold=1.0,                           random\_state=0)  # 运行RANSAC算法  ransac.fit(data, ellipse\_model)  # 计算内点数目和外点数目  inliers = ransac.inlier\_mask\_  outliers = np.logical\_not(inliers)  # 重新拟合椭圆  a, b, h, k = np.linalg.lstsq(data[inliers], ellipse\_model(data[inliers], \*ransac.estimator\_.coef\_), rcond=None)[0]  # 输出拟合结果  print('a =', a)  print('b =', b)  print('h =', h)  print('k =', k)  可以参考一下，chatgpt给的代码，可能不对 |

### 1.2 论文笔记

#### 1.2.1 精度问题

论文中使用RANSAC对线激光与圆柱相交的点云进行椭圆拟合。使用椭圆估计中心点可能会出现误差。

|  |
| --- |
|  |

## 1.3 论文待解决问题

|  |
| --- |
| 最优化问题求解部分没有仔细看 |

## 2. 论文 - 线激光器的手眼标定方法

### 2.1 四元数 – 含未解决问题

|  |
| --- |
| [四元数——基本概念 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/27471300) – 这是高阶篇，以后可以看看  不理解 |

## 手眼标定论文小结

相同手眼标定方法，其结果精度会存在较大差异，不排除论文数据有问题，从目前两篇文章来看还没有办法得知他们出现精度差别的地方在哪里。

\*\*还有重要的一点，在刚开始接触一项新的技术的时候可以适当的看一些学位论文（综述），之后应该看相关技术的小论文，比较深入了解这项技术之后可以再找一些相关的大论文来看。

# 8.18

## 1.wsl & Docker

|  |
| --- |
| [win10离线安装WSL2 Ubuntu20.04系统 - 简书 (jianshu.com)](https://www.jianshu.com/p/4a1748a0112f)  [docker 桌面版报错error during connect: This error may indicate that the docker daemon is not running.:\_程序工厂的博客-CSDN博客](https://blog.csdn.net/tangcv/article/details/112238084) |

# 工作安排(8.21-8.25)

论文，点云代码（提取，特征匹配，特征点提取 - 可以补充一些代码）

论文看完之后要总结，尝试性的看一些外文文献。月度汇报可以再更改总结一下。

# 8.21

## 1.最小二乘法

|  |
| --- |
| [最小二乘法（least sqaure method） - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/38128785) |

# 8.22

## 1.手眼标定相关知识

### 1.1 欧拉角-四元数-旋转矩阵

|  |
| --- |
| [欧拉角和旋转矩阵之间的转换 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/144032401) – 有欧拉角转旋转矩阵的代码  [旋转矩阵（Rotation Matrix)的推导 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/433389563) |

# 8.23

## 1. 点云 – 网络

|  |
| --- |
| [(36 封私信 / 80 条消息) 点云数据可以用来干什么？ - 知乎 (zhihu.com)](https://www.zhihu.com/question/432356705/answer/3162669329) |

# 8.28

## 1. 手眼标定相关博客

|  |
| --- |
| [关于手眼标定的误差计算\_手眼标定误差计算\_boss-dog的博客-CSDN博客](https://blog.csdn.net/qq_45445740/article/details/123573552?ydreferer=aHR0cHM6Ly9jbi5iaW5nLmNvbS8%3D?ydreferer=aHR0cHM6Ly9jbi5iaW5nLmNvbS8%3D?ydreferer=aHR0cHM6Ly9jbi5iaW5nLmNvbS8%3D)  [三维旋转：欧拉角、四元数、旋转矩阵、轴角之间的转换 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/45404840)  [（一）关于手眼标定理论相关的笔记\_boss-dog的博客-CSDN博客](https://blog.csdn.net/qq_45445740/article/details/122170029)  [（二）2D视觉机器人的手眼标定流程记录\_2d手眼标定动作\_boss-dog的博客-CSDN博客](https://blog.csdn.net/qq_45445740/article/details/123947559?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522169319042116800192270979%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=169319042116800192270979&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-2-123947559-null-null.268%5ev1%5ekoosearch&utm_term=%EF%BC%88%E4%BA%8C%EF%BC%89&spm=1018.2226.3001.4450)  [（三）手眼标定结果的应用\_手眼标定的结果\_boss-dog的博客-CSDN博客](https://blog.csdn.net/qq_45445740/article/details/123567627)  [SLAM经典文献之：Hand-Eye Calibration（手眼标定） - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/382606368)  [OpenCV手眼标定（calibrateHandeye()）\_hellohake的博客-CSDN博客](https://blog.csdn.net/hellohake/article/details/104808149) |

# 8.30

## 1. vs问题 – 无法Debug

|  |
| --- |
| 按照7.17-2.3重新配置即可 |

## 2. vs问题 – 使用pcl智能指针报错 – 错误代码-1073740940

|  |
| --- |
| [PCL求助（一）-- 指针释放\_pcl中ptr释放问题\_看到我请叫我学C++的博客-CSDN博客](https://blog.csdn.net/qq_45006390/article/details/119645441) VS可以通过项目属性->C/C++->代码生成->启动增强指令集->选择AVX来解决，我用的是PCL1.12.1版本 |

# 8.31

## 1. PointNet – 包含论文解读

|  |
| --- |
| [最全PointNet和PointNet++要点梳理总结\_X\_Imagine的博客-CSDN博客](https://blog.csdn.net/kxh123456/article/details/121105666) |

# 9.1

## 1.关于点云提取

|  |
| --- |
| 不要再随便调参数了，仔细看看法向量计算原理，边界点计算原理。  PCL智能指针：直接将指针传入函数，函数中使用指针就会对指针进行修改，不需要返回指针。经过更改之后，代码运行速度搜搜快，也不会出现报错问题。 |

## 提取成功

|  |
| --- |
| 终于提取成功了，按照之前的想法，循环即便流程。  但是程序还有待改进，短边滤的有点窄，且右边有一点被截断了  但是这个莫名其妙，这个目标点云(短边)在代码中被认定为边界点 - 参考长边法线角度阈值 |
|  |
|  |
|  |

## 点云提取截图

|  |  |
| --- | --- |
|  |  |

# 9.15

## 1. 手眼标定相关博客

|  |
| --- |
| [机械臂手眼标定原理及代码\_右乘联体左乘基\_绿竹巷人的博客-CSDN博客](https://blog.csdn.net/weixin_42156097/article/details/109506616) |

# 10.8

## 1.失败 - 搭建socks5代理服务器

|  |
| --- |
| [服务器搭建Socks5代理详细教程-帮助文档-蓝米云 (lanmicloud.com)](https://www.lanmicloud.com/doc/480.html)  搭建不了啊 |
|  |

## 2. 突然无法使用conda命令

|  |
| --- |
| 在命令行无法使用conda命令，但是conda --no-plugins可以，这个命令表示的是禁用所有插件，然后就可以使用了。之后使用conda --no-plugins update conda更新conda即可，之后conda命令就可以用了。 |

# 10.10

## 1.关于访问某些网页很慢的问题

|  |
| --- |
| 1. 打开网页控制台，查看哪些域名地址访问过慢，使用ip测速工具找到一个访问比较快的站点，然后更改C:\Windows\System32\drivers\etc\hosts文件 |

# 10.18

## 求解手眼矩阵(TSAI & PSO)

|  |
| --- |
| 结论，现有数据差别不大。其次c++速度比python快的不止一个量级。 x,y,z误差的平均：TSAI在0.24多一点，PSO也就在0.23多一点。 |
|  |

## 胶道点云中心线提取

|  |
| --- |
| 根据每条线激光对应点云中点的y轴坐标相同，求解x轴中值并提取对应点即可。 |
|  |
|  |

## FRP实现内网穿透

|  |
| --- |
| [5 分钟，使用内网穿透快速实现远程桌面 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/410987144) |
| [Frp内网穿透-CSDN博客](https://blog.csdn.net/weixin_43487532/article/details/124266074) |

## 使用TSAI和PSO分别求解手眼矩阵

|  |
| --- |
| 近期工作主要是对网上找到的一些关于手眼标定的数据进行求解。使用经典的TSAI方法以及粒子群最优化算法。以下是针对用一组数据求解手眼矩阵得出的误差分析，就目前的实验来看，TSAI和PSO算法的最终误差相差不大，后续如果针对具体细节对PSO算法进行更进一步的改进，也许会较之现在有明显的提升，此外因为PSO算法是求解最优化问题的算法，对于最后的手眼标定矩阵的解并无唯一，相反，TSAI是专门求解手眼矩阵的算法，最后的求解总是相同的，后续可能需要考虑这方面的因素。­­  TSAI    PSO(没有做限制条件)    粗限制 |

# 10.27

## Wsl配置深度学习

|  |
| --- |
| [2023最新WSL搭建深度学习平台教程（适用于Docker-gpu、tensorflow-gpu、pytorch-gpu) - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/621142457) |

# 10.31

## 1. vscode配置多个分支(实现多存储库)

|  |
| --- |
| [使用VSCode便捷实现Git进阶功能!(包含同一项目配置多个远程Git仓库解决方案) ╰( ´・ω・)つ──☆✿✿✿ 项目协同\_vscode管理多个推送地址-CSDN博客](https://blog.csdn.net/qq_39611230/article/details/111570527) |

## 2. 前端预览DOCX文件

|  |
| --- |
| [mammoth.js 预览 word docx 文档 使用示例 demo example](https://jstool.gitlab.io/zh-cn/demo/preview-ms-word-docx-document-in-browser/) |