# robotCV-pipeline

宋运周 3190101087

## 流程示意图

slight move:

optical flow

classify

&

bounding box

strong move:

essential matrix

fine-Granularity mask

input image

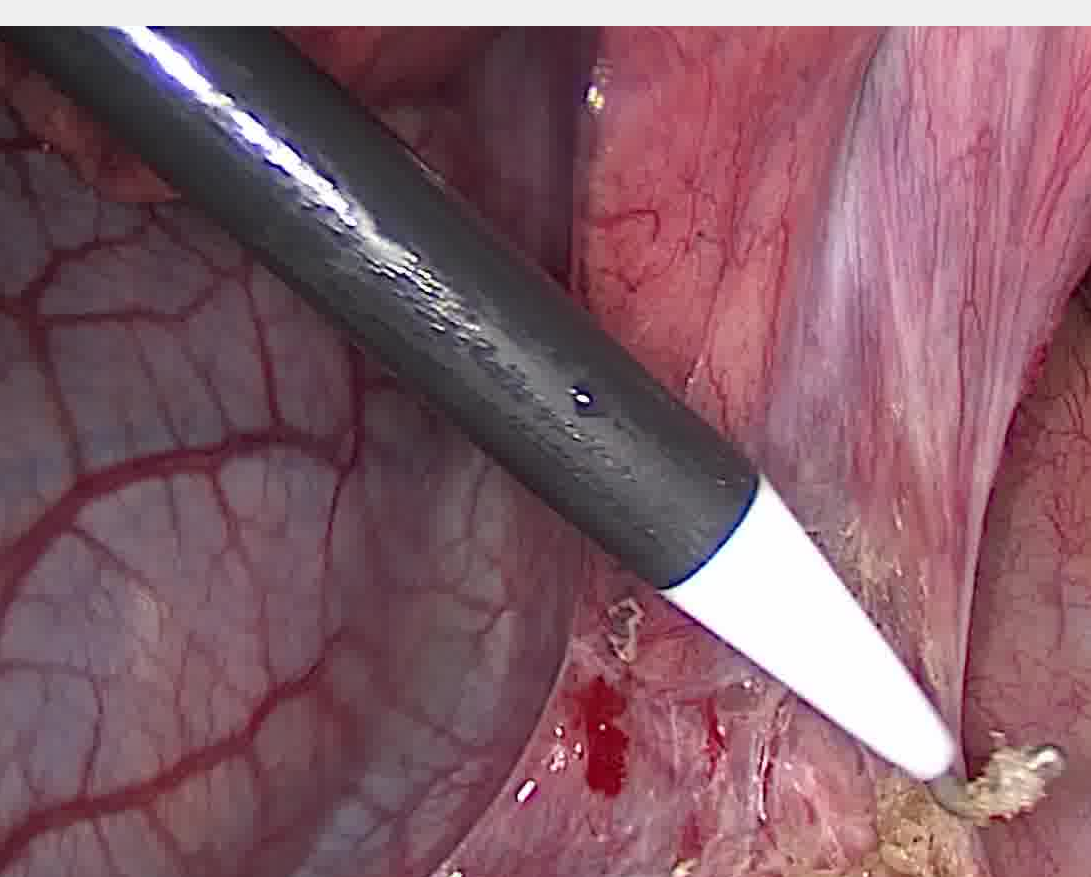
## classify

目前采用faster-rcnn，训练、预测效果正常，但接口使用稍有不当导致耗时较高。日后考虑换成YOLOv3进行预测。

## fine-granularity mask

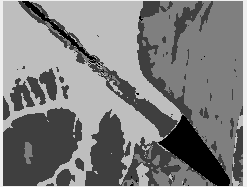
在已知背景、前景区域的情况下为了更好地定位前景器械，对bounding-box内部进行mask细分。具体步骤如下：

1. 如果输入区域较大，先压缩图片进行后续操作后还原。此步骤一可以加快后续计算，二可以减少噪点。



bounding-box 原图

1. 双边滤波，保留边界，平滑内部区域。
2. 使用kmeans大致提取主要部分components。



kmeans提取的components

1. 计算bounding-box外像素点平均值作为typical pixel。
2. 对bounding-box内的点计算与外部typical pixel的夹角，夹角较小则视为背景，较大则视为前景。同时将高光点也设为背景。得到初步分割结果origin\_mask。



origin\_mask，白色为前景，黑色为背景

1. 假设origin\_mask为正确的分割，计算kmeans提取的各components的precision 、recall等参数，综合考虑后选择最佳的1~2个components作为前景，剩余都作为背景得到kmeans\_mask。将其与origin\_mask相乘叠加得到raw\_mask。

kmeans-components recall 、precision计算公式如下



选出的kmans\_mask



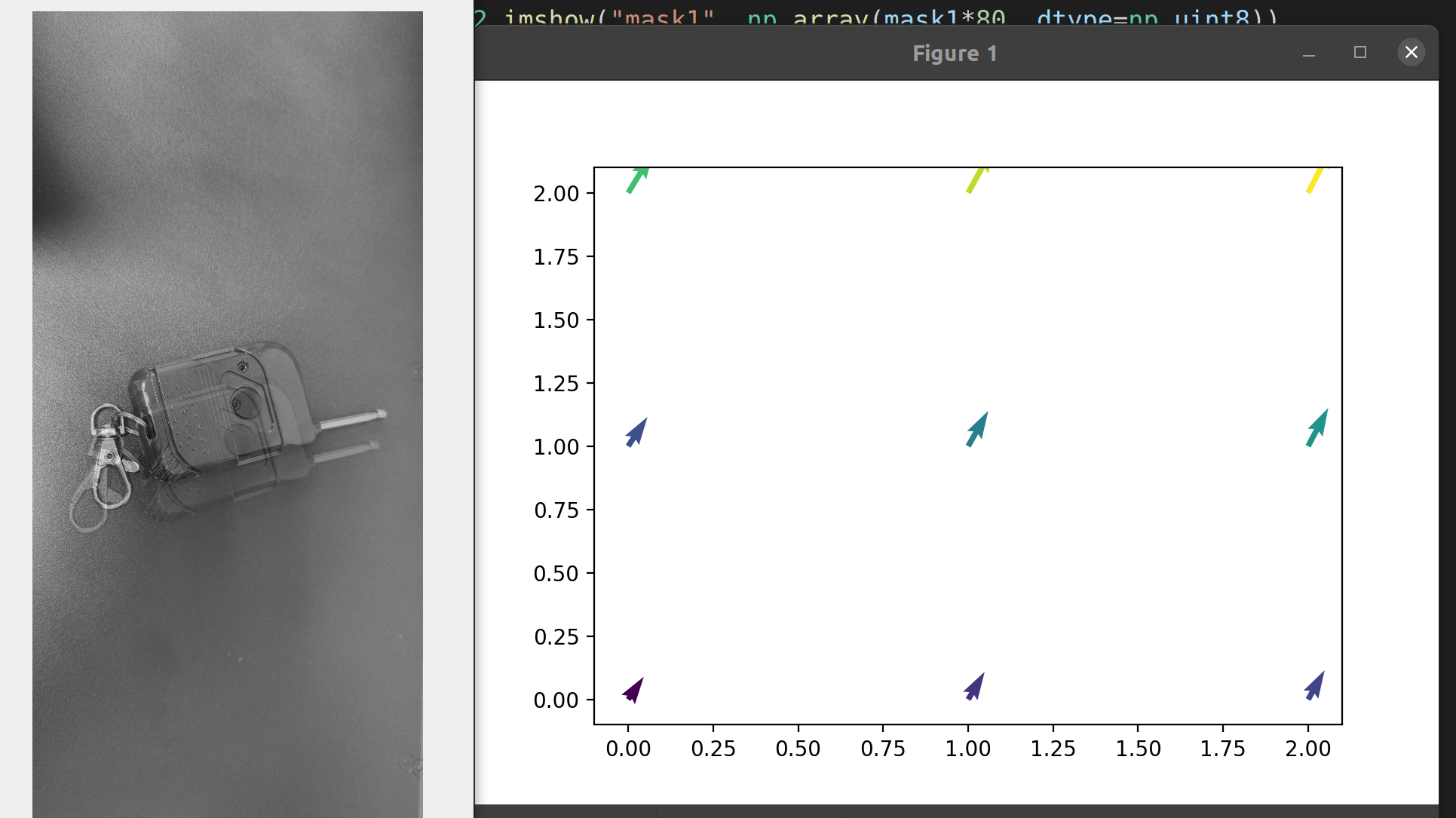
将kmeans\_mask叠加到origin\_mask上得到的raw\_mask

1. 检测raw\_mask中的连通区域connectedComponents，将面积较小的部分都作为背景，得到最终的分割mask。



## optical flow

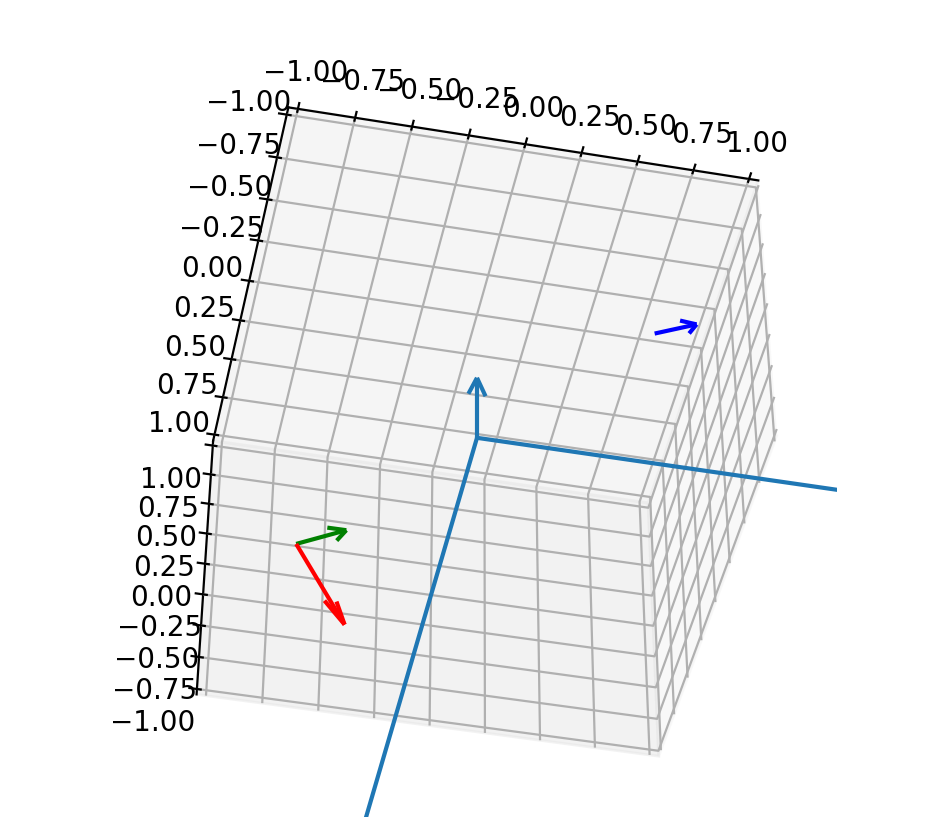
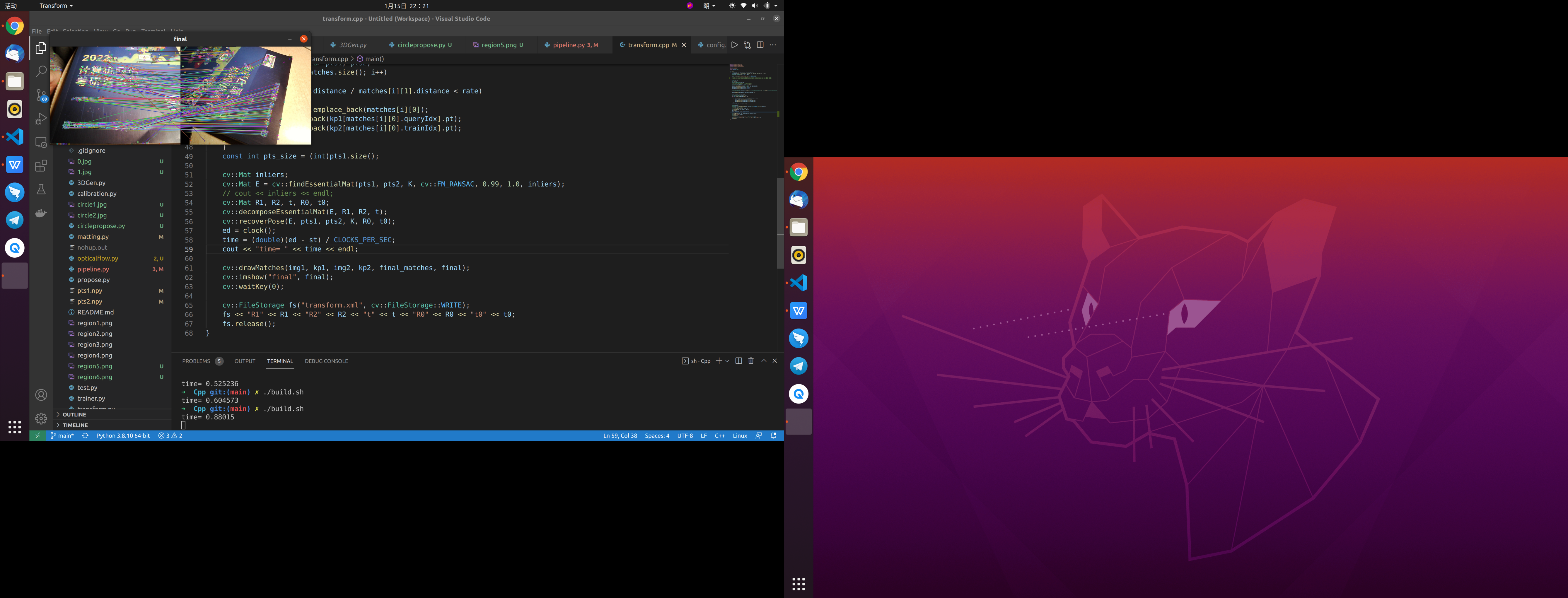
在获得目标器械较为精确的区域mask后，对于幅度较小的运动采用optical flow计算二维位移。即对前景部分的flow进行平均。



如图，对于左图从浅（前一帧）到深（后一帧）计算得到右图所示的位移向量。

## essential matrix

对于幅度较大的运动采用essential matrix估计三维运动变换。essential matrix的解算需要对相机进行标定calibration，这一步骤一开始给我们造成了很大麻烦。由于opencv-python的标定方法存在较大误差导致难以精确标定。最后使用matlab自带的工具箱完成了标定。通过sift特征点提取与匹配可以得到如下变换。



如图，对于上图中从左图到右图的视角变换，解算出下图所示的变换，其中浅蓝色座标系为初始（左图）视角，绿色为右图视角。

## TODO

进程通信

更换YOLOv3或v4，训练模型

多线程调度

切换C++/CUDA加速