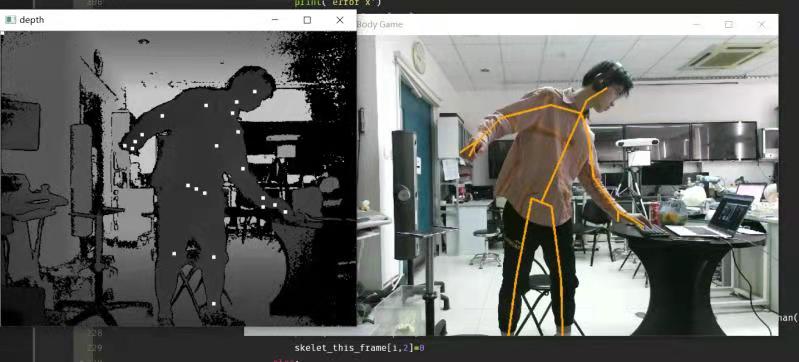
12周进度报告

李浩伟

初版代码用pygame作为界面运行

解决了ctype接口的问题，转换器在输入溢出时会出现不可except的错误，在不报错的情况下强制推出线程，预测骨架点有在rgb点外的情况，已经测试过在深度图像上的追踪点跟随情况，效果可以接受。



准备了多线程运行，目前有多个线程分别进行数据获取，数据处理，数据储存，数据外发，数据清理等不同作用。pygame在使用多线程时在200fps左右位置出现卡顿和无响应，控件反应失效。怀疑其底层线程实现和采用的多线程阻塞。因此去除所有pygame显示，增加cv2线程，用imshow进行显示，原先用pygame的clock控制帧率，现改为time.sleep强制睡眠，测试发现帧率可以超过60，在1/80 sleep的条件下几乎可以跑满80fps，接口速度和效率极高。

通过clean\_up线程处理了内存溢出的问题，总rgb储存在超过2000fps，数据量2000f\*1920pixel\*1080pixel\*4path时发生数据溢出(>16G)，系统崩溃。强制设定最高储存为200f，此时内存占用<1.5G。

经估测，笔记本难以完全支持全套运算，经过系统考虑，将数据上发服务器进行数据处理。发送包为mat，建立两个socket信道，包裹内含发送时间戳，上行数据，下行结果。

联合服务器系统调试，成功运行。有回收输出结果，下一阶段计划进行统计。网络需要针对应用场景进行进一步修改。cv显示中应当添加结果的输出。

统计有两种解决方案，一个是在服务器端统计，一个在本机统计。目前打算在本机上用简单的方法实现一些统计功能，将下发的结果进行储存，而后另开线程进行统计，产生输出，这部分输出同样需要做在CV上。

现有问题：程序在某些情况下在运行一段时间之后（一般大于1分钟）后会发生不可逆的自动弹出，目测是哪个映射器的数据越界了。但是只有在一些奇怪的条件下才会触发，而且在节点图上并不能看出效果。但是崩溃的问题不解决似乎统计也就失去了意义。

这个放在最后修复，现在好歹架子跑起来了。

下周这部分要考虑检测xyz重建算法的正确性，找几个结果数据看一下。

视频心率

打算现在笔记本上实现再考虑迁移到Kinect上。

用笔记本摄像头编写录像代码，收集rgb信号，采用多线程实时绘图显示。

收集一段1000frame的信号

以下是小波分解的结果，视频帧率为30fps，在d5里有明显心率痕迹。

下周进行实时滤波和心率输出，以及程序迁移。

