

Pi-aSLAM 研究记录

1 主动探索

1.1 手控+SLAM 测试

1.1.1 测试流程

将小车切换到车载电源后连接电脑。

为了方便，可以写一个usb_cam.launch，内容为usb_cam-test.launch去掉image_view的部分。

为了方便起见，可以写一个launch文件来启动所有的节点

```
<launch>
  <arg name="path_to_vocabulary" default="/home/pi/Pi-aSLAM/ORB_SLAM2-
master/Vocabulary/ORBvoc.txt"/>
  <arg name="path_to_settings" default="/home/pi/Pi-aSLAM/ORB_SLAM2-
master/Examples/ROS/ORB_SLAM2/Asus.yaml"/>
  <!-- launch usb_cam -->
  <include file="$(find usb_cam)/launch/usb_cam.launch"/>

  <!-- launch ORB-SLAM2 mono -->
  <node name="Mono" pkg="ORB_SLAM2" type="Mono"
    args="$(arg path_to_vocabulary) $(arg path_to_settings)" respawn="true"
    output="screen">
  </node>
</launch>
```

打开一个终端窗口，运行launch文件

```
roslaunch ORB_SLAM2 ros_mono.launch.launch
```

打开另一个窗口，运行手控程序

```
cd ~/ros_catkin_ws
bash src/dumbpi/test_move.sh
```

此时rosnode list应该能看到5个node

```
/Mono
/dumbpi_controller
/dumbpi_keyboard
```

```
/rosout
/usb_cam
```

分别是：单目SLAM，小车驱动，键盘输入，ROS系统输出，相机

然后就可以手控小车SLAM了。

测试完成后，退出roscore，然后关机，然后关闭小车电源。

如果退出正常，应该能在ORB-SLAM2-master/文件夹下看到KeyFrameTrajectory.txt文件。该文件保存了最终计算的相机轨迹，每行分别是：Unix时间戳、3D位置坐标、四元数位姿坐标。

1.1.2 测试获得的信息

1. 初始化需要找一个纹理丰富的区域，小车距离目标约1m，手控小车进行平移，使小车位移1-2个车身长度，然后将摄像头微调至初始目标方向，即可成功初始化。（即：单目初始化必须有平移，不能只有纯旋转。）
2. 单目SLAM的跟踪十分依赖于视野的连续性。以目前的处理速度（1-2FPS）和小车运动速度（500-1000cm/s），小车运动过程中几乎不可能跟踪的上，因此必须使用走走停停策略。
3. 距离障碍物距离小于70cm左右时，转视角会导致视野大幅变化，导致丢失。丢失后视角转回去一般就可以重定位到。
4. 单目SLAM非常依赖纹理和光照。暗处、纹理单一的墙面或家具几乎无法识别特征点。建议使用纹理丰富、颜色鲜艳的各类障碍物。
5. 回环检测没有明确看到，但小车转一圈以后可以识别出一开始的特征点。
6. 比较远的特征点由于没有视差，一般距离估计都很离谱。

1.2 主动探索方案

1.2.1 基础方案

近距观察+拉远视野

特征地图转换为2D格点地图

从格点地图估计前方障碍物远近

获取当前帧可见特征点个数

1.2.2 编译流程

首先要将新编写的 Map2d.cc 和 Controller.cc 加入 CMakeLists。把WiringPi加入link library列表。注意主目录和Example/ROS里的都要加。

为了方便，可以写一个build_light.sh，把编译第三方库和解压dictionary部分去掉，节省时间。此外，可以去掉makefile里非单目的部分。

然后，扩大内存交换区，防止内存不足

```
sudo nano /etc/dphys-swapfile
```

编辑 CONF_SWAPSIZE 变量：从100增加到2048

重新启动交换服务:

```
sudo /etc/init.d/dphys-swapfile stop
sudo /etc/init.d/dphys-swapfile start
```

然后，进入ORB-SLAM2主目录，开始编译. 注意ROS部分的可执行文件是分开的。

```
./build_light.sh 2>&1
./build_ros.sh 2>&1
```

编译成功后，及时调整交换区大小，避免闪存卡老化。

```
sudo nano /etc/dphys-swapfile
```

编辑 CONF_SWAPSIZE 变量：从2048恢复到100

重新启动交换服务:

```
sudo /etc/init.d/dphys-swapfile stop
sudo /etc/init.d/dphys-swapfile start
```

1.2.3 测试流程

为了方便使用root权限，可以写一个启动脚本Pi-aSLAM/start_slam.sh

```
source /home/pi/ros_catkin_ws/devel/setup.bash
cd /home/pi/Pi-aSLAM/ORB_SLAM2-master/
roslaunch ORB_SLAM2 ros_mono.launch
```

首先打开终端，运行脚本。注意此处**必须使用root权限**，否则驱动无法工作。

```
sudo su -c ../start_slam.sh
```

此时rosnode list应该能看到3个node

```
/Mono
/roscout
/usb_cam
```

[BUG.1] 新写的Controller线程不输出

查看日志发现系统其他线程均输出正常。实际原因：sleep函数在linux上以秒而不是毫秒为单位。

[BUG.2] Controller控制车轮不转 实际原因：sleep只接受整数输入，毫秒级别延迟需使用msleep。该函数需要添加unistd.h头文件。

[Bug.3] 重建2D地图时闪退bug 问题原因：非重建置零时使用错误的矩阵清零方式，导致长周期指针指向短周期对象，销毁后内存泄漏。修改后恢复正常。

1.2.4 录制视频流程

1. 启动程序，等待加载ORB词典
2. 开始SLAM后立刻调整图像框位置
3. 电脑先切vscode再切VNC，win+G开始录屏
4. 开始手持手机录像

1.2.5 VO全流程伪代码

```
 $Y_{map} = \text{\textcolor{red}{empty}}$   $X_{trail} = x_0$   $FeaturePoints_{unmatched} = \text{\textcolor{red}{empty}}$   $Frame_0 = get\_frame()$   

 $FeaturePoints_0 = get\_feature\_points(Frame_0)$ 
```

```
$while(i \neq N) 🤔  $Frame_i = get\_frame()$   $FeaturePoints_i = get\_feature\_points(Frame_i)$   

$\quad$ if( $Y_{map} \neq \text{\textcolor{red}{empty}}$ ) 🤔  $Z_{observable,i} = match(FeaturePoints_i, Y_{map})$   

 $Y_{observable,i} = find\_corresponding\_points(Z_{observable,i})$   

 $X_{related,i} = find\_corresponding\_positions(Y_{observable,i})$   

 $x_i = minimize\_remapping\_error(X_{related,i}, Y_{observable,i}, Z_{observable,i})$   $else :$   

 $x_i = Initialize(FeaturePoints_0, FeaturePoints_i)$   $X_{trail}.add(x_i)$   

 $PairsOfFeaturePoints_i = match(FeaturePoints_{unmatched}, FeaturePoints_i)$   

 $Y_{matched} = Triangle\_Measure(PairsOfFeaturePoints_i)$   $Y_{map}.add(Y_{matched})$   

 $i = i + 1$ 
```

1.3 2D地图调试

[BUG.4] 障碍判定不鲁棒，容易被错误特征点迷惑导致障碍过近 问题原因：经初步分析，至少有如下几个问题

1. 目前的障碍仅仅基于特征点的投影，并没有进一步进行滤波，无法排除干扰
2. 目前的投影并不是按照严格水平面投影，而是和真实平面有一个夹角，导致地图不准
3. 目前障碍仅考虑相机镜头左右 $\pm 15^\circ$ 的障碍点，范围不够大

解决方法：首先增大搜索范围。其次，考虑使用二值运算进行滤波。

2 主动重访

3.1 获取地图点的BA误差

在LocalMapping线程中，修改了调用局部BA的逻辑，使BA后保留optimizer对象。从该对象中优化图的边中即可获得每个被优化地图点的BA误差。

3.1.1 重编译g2o库

为了获取图优化后每条边的误差，修改了g2o库的头文件。修改完需要重新编译g2o库

在build_light.sh的开头加入：

```
echo "Configuring and building Thirdparty/g2o ..."  
cd Thirdparty/g2o  
mkdir build  
cd build  
cmake .. -DCMAKE_BUILD_TYPE=Release  
make -j4  
cd ../../..
```

然后运行build_light.sh 即可重新编译g2o

3.1.2 其他错误

[BUG.5] Map2d.h包含eigen头文件出错。

```
fatal error: Eigen/core: No such file or directory  
#include "Eigen/core"
```

解决方法：大小写出错，应为 #include "Eigen/Core"

[BUG.6] 未知原因闪退

问题定位：获取误差平方后尝试调用鲁棒核函数，但边可能未设置核函数 解决方法：判断核函数是否设置，若设置则调用，否则直接使用平方值 是否解决：是

3.2 建立error map

3.2.1 重定义Map2d数据结构

将原有基于opencv的Mat的数据结构修改为基于结构体二维数组的结构，使用更加简单。

[BUG.7.1] 未知原因崩溃

报错如下，或无报错闪退。

```
OpenCV(3.4.3) /home/pi/Downloads/opencv-3.4.3/modules/core/src/matrix.cpp:405:  
error: (-215:Assertion failed) m.dims >= 2 in function 'Mat'
```

问题原因：新建map时没有给成员变量赋值而是给一个局部变量赋了值，导致成员变量未初始化。解决方法：改成给成员变量赋值 是否解决：部分

3.2.2 添加障碍点滤波函数

[BUG.7.2] 障碍点滤波过程中未知原因崩溃 问题定位：生成二值图像函数矩阵维度问题。问题原因：cv::Mat::zeros参数顺序是行数、列数、数据类型。要使(0,0)在图像左下角，行数应为z方向范围，列数应为x方向范围。同时使用at<>访问元素，形参顺序是行号、列号。行号应为z方向范围-z方向索引-1，列号应为x方向索引。是否解决：部分

[BUG.7.3] 障碍点滤波过程中报错崩溃

```
OpenCV(3.4.3) /home/pi/Downloads/opencv-3.4.3/modules/imgproc/src/connectedcomponents.cpp:3928: error: (-215:Assertion failed) connectivity == 8 || connectivity == 4 in function 'connectedComponents_sub1'
```

问题原因：调用connectedComponentsWithStats函数时少传了connectivity参数 是否解决：部分

[BUG.7.4] 障碍点滤波过程中未知原因崩溃 问题原因：局部cv::Mat类型变量访问元素时使用的at类型参数不对，且行号未-1 是否解决: 是

[BUG.8] 障碍点滤波过程中背景被全部选中 问题原因：选取连通域时未排除最大连通域 解决方法：判断连通域外接矩形和全图大小关系，若大小接近则不选中 是否解决：是

3.2.3 error map算法

基于高斯模糊和特征点error值，进行error map计算。首先将每个特征点的error赋值到其所在的2D grid，多个特征点则叠加。其次，进行高斯模糊。

[BUG.9.1] error map更新时未知原因崩溃 问题定位：在地图还未初始化时local mapping线程就尝试更新error 解决方法：local mapping线程判断2dmap初始化情况后再尝试更新 是否解决：部分

[BUG.9.2] error map更新时未知原因崩溃 问题定位：2d map更新周期太长，尝试更新error时，2d地图和3d地图不匹配 解决方法：controller线程减小更新地图周期 是否解决：部分

[BUG.9.3] error map更新时未知原因崩溃，报错如下

```
OpenCV(3.4.3) /home/pi/Downloads/opencv-3.4.3/modules/core/src/matrix.cpp:405: error: (-215:Assertion failed) m.dims >= 2 in function 'Mat'
```

问题定位：某处调用cv::Mat复制构造函数时，传入对象的维度数量小于2 问题定位：Controller线程调用UpdateType时产生了该opencv错误，矩阵维数不对。问题定位：获取相机位姿时，根据当前帧的Tcw矩阵获取，该矩阵可能还未赋值，因此维数不满足要求。解决方法：判定该矩阵行列是否满足要求，若不满足则不更新相机。是否解决：是

[BUG.10.1] error map更新时, 坐标点超出地图范围 问题定位: 2d map更新周期太长, 尝试更新error时, 2d地图和3d地图不匹配 解决方法: controller线程减小更新地图周期 是否解决: 部分

[BUG.10.2] error map更新时, 坐标点超出地图范围 问题定位: 坐标转换出现问题, Vec3d到cv::Mat再到Vec2d出错 解决方法: 更改坐标转换方法 是否解决: 部分

[BUG.10.3] error map更新时, 坐标点超出地图范围 问题定位: 2d map更新不及时 解决方法: 在error map更新前强制更新2d map 是否解决: 是

[BUG.11] 生成error地图时报错如下

```
OpenCV(3.4.3) /home/pi/Downloads/opencv-3.4.3/modules/imgproc/src/smooth.cpp:3820:
error: (-215:Assertion failed) ksize.width > 0 && ksize.width % 2 == 1 &&
ksize.height > 0 && ksize.height % 2 == 1 in function 'createGaussianKernels'
```

问题原因: 高斯核大小必须是奇数(5,5)不能是偶数(4,4) 是否解决: 是

[BUG.12] 出现throw const char*型闪退

```
terminate called after throwing an instance of 'char const*'
[Mono-3] process has died [pid 12584, exit code -6,
```

问题定位: 获取误差地图 GetErrorMapImage 时已有2D地图上没有误差。 问题定位: GetErrorMapImage() <- GetErrorMapImageColor() <- UpdateError() <- Update2dMap() 或 GetErrorMapImage() <- UpdateError() <- Update2dMap() 即LocalMapping线程试图绘图时, 地图上没有误差 问题原因: UpdateError在更新完成后绘图前释放了互斥量, 导致Controller线程重新更新地图, 抹掉了所有的error 解决方法: 1: 不释放互斥量直到绘图结束 解决方法: 2: 重写2D地图更新逻辑, 更新地图不影响之前的error

3.2.4 重写2D地图更新逻辑

2D地图应当遵守增量更新的规则。

- 地图大小和占用情况更新逻辑
 - 根据当前3维地图特征点的x和z范围, 计算新的2D坐标范围
 - 分配新的2D地图空间
 - 将原坐标范围内的误差数据原样复制到新范围的对应范围内, 占用数据不复制
 - 删除原数据空间, 更新数据索引和坐标范围索引
 - 更新占用数据
- 误差更新逻辑
 - 获取所有待更新误差的特征点及其2D坐标范围
 - 建立临时误差图, 将特征点误差累计到临时特征图上
 - 用临时误差图更新2D坐标范围内的2D地图
 - 其中临时误差图误差为0的格点不更新

[BUG.13] 编译问题

```
/home/pi/Pi-aSLAM/ORB_SLAM2-master/src/Map2d.cc:384:19: error: request for member
'find' in 'ORB_SLAM2::posMap', which is of non-class type 'ORB_SLAM2::PosMap()'
{aka 'std::unordered_map<g2o::OptimizableGraph::Edge*, Eigen::Matrix<double, 2, 1>
>()'}
```

```
    if(posMap.find(e) == posMap.end()){
        ^~~~
```

问题原因：无参构造类的成员时不应该加括号 PosMap posMap(); -> PosMap posMap; 是否解决：是

[BUG.14.1] 未知原因无报错崩溃

```
[Mono-3] process has died [pid 14562, exit code -11,cmd /home/pi/Pi-
aSLAM/ORB_SLAM2-master/Examples/ROS/ORB_SLAM2/Mono.....
```

问题定位：地图大小更新时，混淆了旧地图大小和新地图大小，造成数组越界。解决方法：清晰定义旧大小和新大小 是否解决：部分

[BUG.14.2] 未知原因无报错崩溃

```
[Mono-3] process has died [pid 14562, exit code -11,cmd /home/pi/Pi-
aSLAM/ORB_SLAM2-master/Examples/ROS/ORB_SLAM2/Mono.....
```

问题定位：地图大小更新时，新地图大小比旧地图小，数据无法迁移 解决方法：若新地图小，则不进行迁移 是否解决：部分

[BUG.14.3] 无报错崩溃

```
[Mono-3] process has died [pid 14562, exit code -11,cmd /home/pi/Pi-
aSLAM/ORB_SLAM2-master/Examples/ROS/ORB_SLAM2/Mono.....
```

问题定位：新地图x和z一个维度小，另一个维度大，出现问题 解决方法：新地图大小的各个边界都设置为观测的边界和原边界的极大值 是否解决：是

[BUG.15] Opencv报错

```
OpenCV(3.4.3) /home/pi/Downloads/opencv-
3.4.3/modules/imgproc/src/colormap.cpp:516: error: (-5:Bad argument) cv::ColorMap
only supports source images of type CV_8UC1 or CV_8UC3 in function 'operator()'
```

问题原因：colormap只能对uchar型图片上色，不能对double型上色 解决方法：转换为uchar型 是否解决：是

[TODO] 障碍滤波后应当重新调整地图尺寸 问题：障碍滤波去掉了孤立的特征点，这些特征点不应当被考虑也不应当被计入 优点: 减小2D map大小，节省空间和计算时间 难点：地图范围除了考虑障碍外还要考虑相机位置（后续决定不考虑该问题）

3.2.5 移动系统实验

测试到目前为止完成的所有模块（Controller、2D障碍图维护、2D误差图维护）是否能在真实环境下协调工作。

实验情况：闭环前功能正常，需要一个比较封闭、明亮、纹理丰富的实验环境。闭环暂未测试过。

3.2.6 更改2D地图显示区域

当前显示区域过大，主要信息都在左下角无法拖到中间。更改地图显示逻辑，只显示2D地图内相机和障碍滤波后占据格点的部分。

3.2.7 闭环实验

[BUG.16] 未知原因OpenCV错误

```
OpenCV(3.4.3) /home/pi/Downloads/opencv-3.4.3/modules/core/src/matrix.cpp:423:
error: (-215:Assertion failed) 0 <= _rowRange.start && _rowRange.start <=
_rowRange.end && _rowRange.end <= m.rows in function 'Mat'
```

问题定位：显示map的crop的时候范围超过了图像的范围 解决方法：判断是否超过最大范围，超过则使用最大范围 是否解决：是

实验较为成功，闭环后2D地图和实际地图匹配。

3.3 根据error map路径规划

3.3.1 闭环后error map刷新

闭环后会进行闭环修正和全局BA，此时地图的占用情况会改变，全局BA也会产生新的error map。此时需要刷新error map

[BUG.17] 进程崩溃

```
[Mono-3] process has died [pid 2068, exit code -11, cmd /home/pi/Pi-
aSLAM/ORB_SLAM2-master/Examples/ROS/ORB_SLAM2/Mono /home/pi/Pi-aSLAM/ORB_SLAM2-
master/Vocabulary/ORBvoc.txt /home/pi/Pi-aSLAM/ORB_SLAM2-
master/Examples/ROS/ORB_SLAM2/Asus.yaml __name:=Mono
__log:=/root/.ros/log/f2c3f322-de70-11ec-9e01-e45f014d7739/Mono-3.log].
log file: /root/.ros/log/f2c3f322-de70-11ec-9e01-e45f014d7739/Mono-3*.log
```

问题定位：执行全局BA和闭环线程不是同一个线程。如果让全局BA线程初始化optimizer，该线程运行完毕就会销毁其资源。而闭环线程不会等待该线程完成就继续执行。因此，可能发生两种情况导致崩溃。问题原因：

1.全局BA线程执行完毕后，LoopClosing线程使用optimizer资源，但该资源已被销毁。问题原因：2.全局BA线程还未分配资源时，LoopClosing线程就使用optimizer资源。解决方法：在真正运行全局BA的线程执行error map更新

3.3.2 基于error map的路径规划算法

3.4 路径规划的执行

3 其他主动SLAM

3.1 主动闭环

3.2 主动重定位