说明：重装unbuntu后从零开始复现orb\_slam2的总结：

吴博注释代码：<http://git.oschina.net/paopaoslam/ORB-SLAM2>

PPT网站：<https://wenku.baidu.com/view/8cf70ca7bdeb19e8b8f67c1cfad6195f312be813.html>

其他相关网站：<http://paopaorobot.org/420.html>

解压缩：<https://www.linuxidc.com/Linux/2012-08/68122.htm>

tgz：<https://blog.csdn.net/weixin_40533355/article/details/80473223>

数据库云盘：<https://www.sohu.com/a/219232053_715754>

使用自己总结文件：xzbots和廖雪峰Git summary；Linux and Ubuntu；树莓派：linux常用命令以及vim编辑器（命令图）；视觉SLAM十四讲-高翔

相关文献，总结看周报：

* 初步版本-小改版本-slam2：ORB-SLAM Tracking and Mapping Recognizable；ORB-SLAM A Versatile and Accurate Monocular SLAM System；ORBSLAM2\_An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras
* Fast角点：(OFAST1\_CORR)Machine\_Learning\_for\_High-Speed\_Corner\_Detection；(OFAST2)Faster and Better\_A Machine Learning approach to corner detection
* 描述子：Brief\_Binary robust independent elementary features
* ORB特征：ORB\_an efficient alternative to SIFT or SURF
* 词袋：Bags of Binary Words for Fast Place Recognition in Image Sequences
* 图优化：g2o\_A General Framework for Graph Optimization
* 闭环检测和重定位方法：Fast Relocalisation and Loop Closing in Keyframe-Based SLAM
* 3D-2D，用于初始化后的跟踪，重定位：EPnP An accurate On solution to the PnP problem
* 在跟踪线程中局部地图优化时用到相似于double window的方法：double window optimisation for constant time visual SLAM
* 闭环检测计算相似变换、pose graph optimization：Scale Drift-Aware Large Scale Monocular SLAM；closed-form solution of absolute orientation using unit quaternions
* H分解出t、R：Motion and structure from motion in a piecewise planar environment
* E的SVD分解：Multiple View Geometry in Computer Vision.

待通过实验解决：

* 不同层会不会提取到相同的特征点？提取到的特征点会不会匹配上？若匹配上是不是意味着不确定在哪看到的点了？
* Searchbybow函数中未考虑多次的匹配的异常点，可以设计函数放于continue之前计算分数
* 跑同样的轨迹有时会丢失有时不会

问题

* 期望特征数与金字塔尺度关系
* 用滑窗提取特征点的作用
* 索引值为何经常初始化为-1，

解决：若后面将其赋值，那么正常索引大于0，若其小于零则说明未进行任何操作

* 相邻两帧图像检查依据，

解决：两帧图像所有特征点角度变化相似，具体看代码

* SVD分解？F、H分解为什么有几种可能的解？初始化时重建原理？
* 跟踪时位姿得计算过程，直接使用g2o计算？
* VSLAM基础（六）————超定线性方程组的解法，搜索这个系列看看
* Household变换

<https://blog.csdn.net/flyfish5/article/details/52315062>

第二遍待搞清问题，不留疑惑的全看懂，如果是自己该怎么写

* 各线程之间的协调，怎么通过操作内存相互协调
* 局部地图、闭环检测、重定位等理论及具体实现方式

解决：看论文

* 理清各类的作用，搞清public、protect等的作用

解决：protect可以被子类继承而private不能

* 如何评价结果？将结果以可视化的方式表现出来
* 注意观察照片缩放前后的区别，缩放使用什么方法比较好
* 金字塔各层比较和存储的方式，只存储匹配的一帧吗？注意factor的使用

解决：代码中按1.2比例缩小图像，对于同一个物体，相当于距离变远。金字塔的缩放相当于摄像头沿光轴平移，之后改变平移量就好。代码的思路是：通过双线性内插缩放图片，这里图片的间隔是一样的，分辨率不一样。然后使用相同的方法（如相同的patch），提取角点和描述子，最后将不同层（缩小）的特征点的坐标放大（乘对应层数即可）。特征点中还保存着对应层数。还需结合计算过程综合考虑

* 跟踪没用过的特征继续保存在关键帧中吗？

解决：对

* 估计初值怎么得来

解决：第一次计算之后都有初值了

* 优化时用的是特征点世界坐标系坐标到相机坐标系之间的位姿变换，故各帧之间不会相互关联？

解决：优化有很多种，有pose-point、pose-pose

* 相机参数校正过程

解决：提取完ORB特征后在使用校正参数校正特征点的坐标，包括四个顶点的坐标

* KD树词的格式，观察各特征之间1,0差别，一位的偏差是否会带来较大影响
* 地图点的描述子取哪个特征帧的描述子

解决：离其他描述子距离最近的描述子

* 参考帧怎么选取？从covisibility graph（只包括关键帧）吗？还是直接用上一帧作为参考帧，丢失之后才从地图上查找？
* 上次插入关键帧与当前决定插入关键帧之间帧怎么处理？论文中track是指匹配点还是map point？包括恢复的map point吗？
* 建立局部地图，怎么计算期望看到某个点的帧数？有个判断条件用到
* 局部地图停止后是否插入关键帧到local队列？
* dmin、dmax怎么算

解决：在mappoint中有一个参考帧专门供其计算，会随着运动不断更新，dmax=1.2\*该帧光心到该路标点距离\*观测所在层尺度（1.2的n次方），dmin=0.8\*dmax/最底层尺度（1.2的7次方）

* 视差的判断
* 在关键帧中出现但词袋中没有的词怎么处理？不考虑他们会带来什么影响？考虑会不会改善效果？
* 最近邻ratio？
* 通过旁边三个点恢复一些3D点的方法，3D融合过程中先匹配在映射吗？（两帧图都是用loop帧的路标点）？
* 尺寸的初始化？

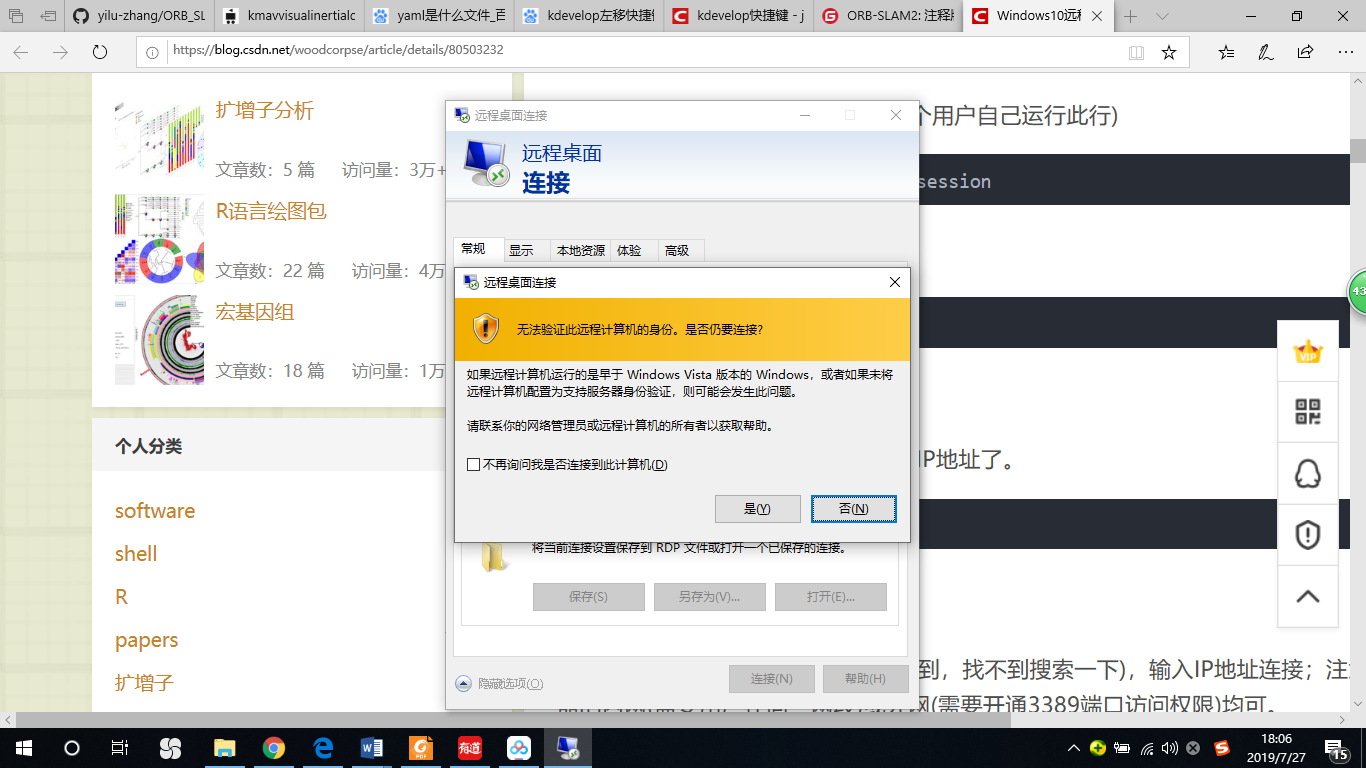
解决：先使t计算位姿时使t的模为1，新建地图时将相机中间深度的距离设为1

* 倒数深度的作用？
* Local队列是在闭环检测的时候用的吗？注意各段时间
* 为什么会有四个解？
* 提取fast角点圆是怎样的？，，
* 只运行tracking线程跑一个循环会不会是运行三个进程时间的3倍？
* 从E分解出R得大于0
* 对于每张图片的特征是离谁最近挂在哪个页节点下吗？id一样？
* EPnP求解控制点坐标：待看代码具体计算过程，22点有参考网站。ICP问题？

1. win10与Ubuntu 16.04远程连接

<https://blog.csdn.net/woodcorpse/article/details/80503232>

仅供参考，最后未成功，IP就是查询到那个。发生如下错误



首先要更新软件源（改镜像）update 和upgrade，装无线驱动

远程桌面连接复制问题：<https://www.cnblogs.com/waw/p/11014100.html>

1. 安装及设置git

安装：sudo apt-get install git -y

设置：<https://www.cnblogs.com/superGG1990/p/6844952.html>

（生成密钥对那就是邮箱，未设置前没有权限下载github上code）

警告：搜索问题，在某个文件后面加上ip即可

1. 下载安装工具及第三方库

参照源码网站：<https://github.com/raulmur/ORB_SLAM2>

包的两种常用方法：

* 直接使用sudo apt-get install 包名 安装
* 到网站下载包含cmakelist.txt文件的压缩包，解压后按cmake的方法安装：在解压目录mkdir build；cd build；cmake ..（根据cmakelist里分析里面文件、相关链接库，生成自动编译脚本Makefile等中间文件）；make –jn //n线程（编译）；sudo make install（安装）

卸载安装包方法：

<https://www.cnblogs.com/blackay03/p/9062503.html>

1. 安装cmake：编译用

sudo apt-get install cmake

1. 安装kdevelop

sudo apt-get install kdevelop

1. 安装vim

sudo apt-get install vim

1. 下载源码

注意按网站上clone语句：git clone https://github.com/raulmur/ORB\_SLAM2.git ORB\_SLAM2，没有后面ORB\_SLAM2会下不了，这与一般的下载不一样

对于DBow2和g2o已经包装在thirdparty文件夹中了，下载好程序后按照github网站上三步运行安装即可：cd ORB\_SLAM2 chmod +x build.sh ./build.sh

安装完上面几项后，用kdevelop打开文件，根据报错信息安装相应包，可参考如下网站

<https://blog.csdn.net/radiantjeral/article/details/82193370>

问题：pangolin使用catkin\_make\_isolated替代catkin\_make

1. 安装opencv（opencv 3.2.0，自己安装，记得先安装依赖项）

<https://blog.csdn.net/ksws0292756/article/details/79511170>

后面发现其与官网步骤类似，最好参考官网步骤：

https://docs.opencv.org/master/d7/d9f/tutorial\_linux\_install.html

注意先解压，解压步骤见最上方网站，然后按网站步骤弄

不先安装依赖项带来的麻烦，及解决步骤

* 报未安装lihgtk2.0-dev 和pkg-config依赖项错

解决：先安装两个依赖项包：sudo apt-get install 依赖项包，使用下网站第一条解决，第一句只是查看当前两个包版本。make比较慢，记得使用多线程。后安装相当于要把依赖项和cmake连接起来，会麻烦一点

<https://stackoverflow.com/questions/28776053/opencv-gtk2-x-error>

* cmake句时报未安装qt4错

解决：sudo apt-get install qt-sdk，然后cmake成功，继续按照上面网站操作就好

1. 安装eigen3（按上面网站）
2. 安装pangolin（按上面网站）

其中src文件夹相当于自己package放文件包的文件夹，不过这里得用src才能执行下面命令，故重新在package目录新建src文件，将pangolin下载此文件夹下。然后按照提示安装catkin，运行catkin\_make出问题后运行提示目录isolate。然后继续走在src上一级会多出build build\_isolated devel devel\_isolated多个目录，下次安装时注意多加一级，这样太乱

1. 1
2. 源码理解

* Tracking线程解析

<https://www.cnblogs.com/wall-e2/p/8057448.html>

1. 各类的作用，首先构建一个system类实例，然后在这个类里面包含所有类的实例指针，有些类可能又在实例指针中再建立实例指针。相机坐标系的中心在光心。自己正面看图片时。代码中横向为x轴-u-cols-width，纵向为y轴-v-rows-height，圆心在左上角。故在用at定位像素是应是<y,x>。new的地方创建一个指针实例。

* Mono\_tum：主函数，在TUM数据集进行单目运算
* System：整个系统相关操作
* Frame：帧，新来的帧，需要在它上面提取特征。包括创建序号mnid和timestamp两个id。对当前帧进行处理，提取特征，供track使用。操作函数在ORBextractor，将提取特征的关键点存于其mvkeys（keypoint）-，描述子存于mDescriptors（mat：cols-特征点个数，rows-描述子32）。用于创建当前帧mcurrentframe。提取特征点后校正特征点x、y坐标，将其存于mvkeysUn（keypoint）。将特征点与mvpmappoints对应起来。

其包含复制构造函数，在初始化时会用于复制类实例

* Keyframe：关键帧，新来的帧满足条件可变为关键帧。初始化时创建初始化地图就有用了，包括创建序号mframeid、mnid和timestamp（与frame相同）3个id。
* Keyframedataset（是system内指针类）：保存所有关键帧
* Mappoint：路标点
* Initializer：初始化时候用，计算H、F矩阵，并分解得出位姿。在tracking中新建该类的指针实例。
* ORBextractor（是tracking内指针类）、Extractornode（不新建指针，只作类型用）：提取特征时用，两个在ORBextractor.h头文件中定义。
* ORBmatcher：特征匹配。在跟踪时新建一个局部类
* Tracking（是system内指针类）：跟踪用，主要在主线程跟踪线程用。包含currentframe

和lastframe用于跟踪。

* Localmapping（是system内指针类）：局部地图构建线程
* Loopclosing（是system内指针类）：闭环检测线程
* Map（是system内指针类）：关键帧和路标点，使用指针，便于节省空间
* Converter：g2o、Eigen、cv之间的矩阵转换
* Optimizer：优化器，BA、pose、sim3、essential graph等优化方法，不见实例，像函数一样调用
* Pnpsolver：3D-2D求解
* Sim3solver：求相似变换

画图用

* Viewer（是system内指针类）
* Plane
* viewerVR
* Framedrawer（是system内指针类）：
* Mapdrawer（是system内指针类）

1. 重要阈值或参数

* covisibility graph：共视个数-；
* 场景识别：最好分数的百分数-；
* DBOW2：K；D-；匹配所选层-；
* 初始化：匹配的特征值最小值-；
* Tracking：金字塔层数-；尺寸因子；欲提取的特征数-；检测FAST的阈值；是否保留某关键点（角点）的规则；局部地图优化K1与当前帧共视帧的个数阈值；及K2帧数阈值；；自适应detector阈值；自适应cell阈值；判断跟踪丢失条件；pnp中inlier数；匹配时方向一致性检测；最近邻ratio-；
* Trackwithmotionmodel（）：位姿初值为mverlocity（之前两帧的运动速度）\*mlastframe，使用lastframe的重投影（重载函数）来找
* Trackreferencekeyframe（）：位姿初值lastframe

跟踪到路标点超过15个。

* 两个都使用pose优化计算pose，同时探测一些异常的路标点，结果>=10返回true
* Tracklocalmap：正常需30个点，重定位阶段需50个点才算成功
* Map：利用旁边N帧进行重建地图；三角化极线约束-；视差；删除帧的百分数-；dmin、dmax；视角差别阈值-60；
* Loop：smin计算边权重；一致性检测；
* EDGE\_THRESHOLD=19：代码中采用101反射方便提取边缘上角点，FAST提取中的6，+-3是为了保证提取的都是图片内的点，而不提取填充上的点。19是为了计算描述子用的，斜框时15\*根号2=21。
* 初始化时：
* 初始化成功要求：当前帧特征点大于100；匹配点大于100；
* 特征匹配：只在金字塔第一层做匹配；在参考帧像素上下左右100个像素点找合适匹配点；汉明距离小于50且最好距离小于次好距离0.9倍；同一个点不能被两次匹配；方向检测词袋数量为30，选择其中三个数量最多词袋，两个次好词袋个数应大于最好词袋的0.1倍，否则丢弃。
* 开辟两个线程计算H和F：从匹配点中随机选出200组（迭代次数）8个点对，特征点可以重复使用，保证每组不重复即可。计算分数阈值看代码和吴博ppt-p24，
* 使用F恢复三维坐标

checkRT：两个在cosparallax<0.99998(即保证不平行)时判断深度是否为负；重投影误差（差的平方和）小于4；保留的点cosparallax<0.99998(即保证不平行)。满足前两个条件就认为是好的点ngood++；满足三个条件才将对应标志置true

选出最好的一组：最好的RT中正确的点要多于局内点的0.9且总数在50以上，其他非正确RT要少于最好的RT点数的0.7倍。

另外parallax（两条射线的角度）要大于1度，当满足要求点多于50个时，要求前50个大于1度，否则要求全部大于1度

* 使用H恢复三维坐标：checkRT和F一样

选明显优于其他组最好的一组（条件为且关系）：最优解点数的0.75倍比次优多；大于局内点的0.9倍且大于50个；视差大于1度

1. 用到opencv类和函数

* OpenCV —数据持久化: FileStorage类的数据存取操作与示例

<https://blog.csdn.net/iracer/article/details/51339377>

* mat: 属性step，size，step1，elemSize，elemSize1。create-直接创建一个矩阵

<https://blog.csdn.net/Felaim/article/details/80702787>

矩阵操作的一些方法

<https://blog.csdn.net/b1342504010/article/details/17717137>

* outputArray：特征提取返回描述子引用时
* copyMakeBorder：扩充边缘：101-以最边缘像素为轴，关于BORDER\_ISOLATED用法在函数声明中有详细描述

<https://www.colabug.com/1691132.html>

* resize：对图像进行放缩，也称图像内插，书P37，内插过程理解：改变放缩后图片像素之间间隔，使其与原图像重合，注意两者间隔不同，放大，间隔变大，缩小间隔变小，在此基础上进行插值。常用双线性插值。

<https://blog.csdn.net/guyuealian/article/details/85097633>

<https://blog.csdn.net/qq_37577735/article/details/80041586>

* 给单通道赋值，注意不同通道的类型

<https://zhidao.baidu.com/question/1050998807787878299.html>

* FAST：threshold为中心点与圆上点边缘差值（本文最大20，最小7），其为FASTx从16个像素选取9个的情况。得到kepoint，得到pt、response可直接用

<https://blog.csdn.net/zhaocj/article/details/40301561>

* Keypoint：pt(0,0)坐标, size(0)代码中PATCH\_SIZE\*缩小比例，描述子尺寸？, angle(-1)方向以x轴到y轴为正方向, response(0)响应强度，和中心点与圆上像素差值有关, octave(0)所在金字塔层, class\_id(-1)

<https://blog.csdn.net/u010821666/article/details/52883580>

* fastAtan2：计算反正切函数，返回结果为0-360，误差0.3
* getMat 、 InputArray 、OutputArray、 CV\_Assert 、 create、\_tchar、wchar\_t、char、typename、类的静态数据成员、Mat

<https://blog.csdn.net/weixin_41484240/article/details/80595507>

* mat.reshape：第一个参数通道数，第二个参数行数.肯定是2\*2

<https://www.cnblogs.com/denny402/p/5035535.html>

* undistortPoints（）：提取完特征点后进行校正

<https://blog.csdn.net/yong_qi2015/article/details/52946821>

* SVDcomp：返回值已经按从大到小排列

<https://blog.csdn.net/weixin_42587961/article/details/96865137>

<https://www.cnblogs.com/mikewolf2002/p/3454760.html>

* norm：求范数
* cvmultransposed：求AT\*A
* cvSVD: <http://www.mamicode.com/info-detail-57033.html>
* cvRealeaseMat: 该函数参数为指向cvmat的图像指针的引用，运行完该函数后使参数指针置空（NULL），该函数与cvReleaseImage(&pImage)的用法类似
* 1

1. std常用函数

* fill：填充数值，<https://blog.csdn.net/cv_jason/article/details/80894886>
* make\_pair:数组对之类，

<https://blog.csdn.net/u011499425/article/details/52756088>

* reshape

<https://blog.csdn.net/u010916338/article/details/84066369>

* pair <int,int> x,x.first即第一个元素，x.second即第二个元素，make\_pair(x[2])将其索引和内容配对
* map<,>：类似于字典，可以像数组一样使用，其下标可以不是int了，另外，一些操作和pair相似。Map.count（key），可以查找是否有下标
* 1

1. g2o

* g2o矩阵使用的是eigen矩阵

1. eigen
2. TUM数据集：包括室内室外范围比较小的场景

* 数据集的文件名就是时间

1. timestamp记录图片的先后顺序，covisibility graph和essential graph 、span tree用于优化
2. 各线程通信标志：tracking用covisibility graph；local mapping会用也会改graph，且操作较频繁；loop会用也会改graph，且改得较少，检测到闭环的时候停止local

* Loop启停local标志位：

1. 文件加下有build.sh，执行了一系列命令行操作，建立好库函数和可执行程序，examples里有各中情形的可执行程序。可以通过cmakelists.txt文件查找生成对应库和可执行程序使用的源文件
2. yaml文件常用于写配置文件，简介如下

<https://www.cnblogs.com/wxmdevelop/p/7341292.html>

1. 使用kedevelop打开吴博注释代码及源码两个工程时，源码system.h的头文件system的构造函数与吴博代码链接上了。原因不详，中间有过挺多乱七八糟操作。如改文件名，build多次
2. 程序的线程之间通过标志位来控制

<https://blog.csdn.net/hzwwpgmwy/article/details/80493247>

1. C++ 互锁mutex的使用，待进一步理解，b站有收藏，lock是保护在其之下的函数及变量？
2. copyMakeBorder：<https://blog.csdn.net/qq_22764813/article/details/52787553>
3. 取整函数：<https://www.cnblogs.com/zjutlitao/p/3558218.html>
4. 立方体可以用八叉树切分，图片可以用四叉树切分
5. 多线程

* 线程名.join:等待直到线程结束

1. 提取fast角点时，在每个cell设定最小数量，从大到小（20-7）设置fast的阈值，直到获得合适数量角点，阈值越大越好
2. 类之间相互包含彼此的指针类，既可以增强彼此的联系，又可以节省空间，使用指针操作相当于在不同类型操作同一片内存，但是多线程操作时可能会出现冲突
3. 计算汉明距离在代码中有快速的计算方法，代码中有相应链接
4. EPnP求解控制点坐标：（实验所得）一个控制点取4个路标点质心坐标，然后去质心后，其余三个控制点取四个路标点PCA主方向上三个点，PCA能更好的表达这四个点的信息

<https://blog.csdn.net/jessecw79/article/details/82945918>

<https://www.jianshu.com/p/312cc524b8d1>

1. 1

