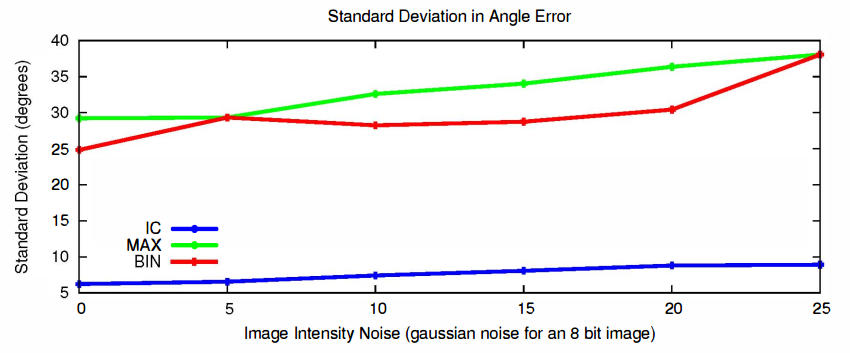
第15周汇报

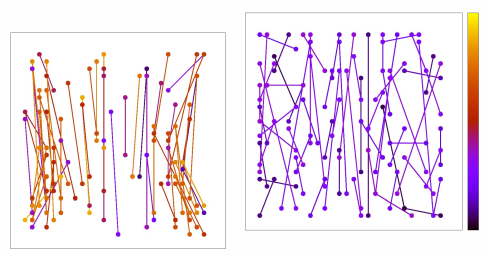
——张溢炉

1. 《ORB: an effcient alternative to SIFT or SURF》总结
2. 内容摘要

本文提出了ORB (Oriented FAST and Rotated BRIEF)特征，有效结合了FAST特征点和BRIEF描述子，并给FSAT加上了方向，从而使BRIEF具有旋转不变性

1. 总结及理解

* 对FAST的改进版：oFAST（Orientation FAST）,其主要有以下几点改进
* 为了解决FAST错误检测到线的问题。本文采取的方法是：先调低阈值获得大于N个角点，再使用Harris corer measure [11]中的方法对角点进行排序取前N个角点
* 为了使FAST获得尺度不变性，本文采取的方法是：在图像金字塔的每层提出角点
* 为了给定FAST角点的方向，本文使用intensit centrid（IC） [22]提出的方法，并将其与BIN和MAX的梯度方法在噪声干扰的的情况下作了比较（结果如下图），证明IC方法具有更好的效果
* 对BRIEF的改进版：rBRIEF，其主要有两个目的，具体步骤在论文有注释

1. 使每位描述子具有高方差，从而使不同关键点描述子之间具有更好的区分性
2. 使描述子各位之间相关性低，从而使各位之间更加独立，下图是通过相关性选择前后（左右）的点对，比较直观的感觉就是左右有较多接近平行且靠得比较近的点

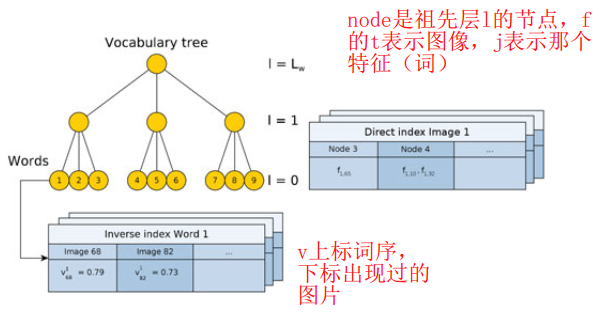
* 从实验结果来看，ORB特征具有较好的旋转不变性。但是使用图像金字塔的方法并没有使其具有较好的尺度不变性，还需改善

1. 《Bags of Binary Words for Fast Place Recognition in Image Sequences》总结
2. 内容摘要

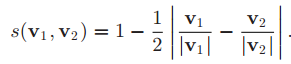
本文提出了一种从一串经过的图片中识别场景（Place Recognition）的新方法，其以FAST+BRIEF为特征，处理速度可达到22ms/帧，比之前的方法快了一个数量级，而且它能保证检测到的都是正确的

1. 总结及理解

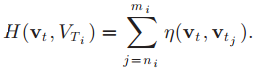
* Place Recognition对回环检测和重定位非常重要
* 存储图像描述子的树如下图，其中words由所有描述子的特征向量组成，direct index用于由图片查找其对应特征，inverse index用于由特征查找出现过的图片。先使用训练集图片获得足够大的树，然后再在运行更新direct index和inverse index。假如有M个特征（words），先使用K-means++将其聚成k类，然后再根据设定树的深度Lw，继续聚类。这样在查询特征时可以将时间复杂度变为log级。



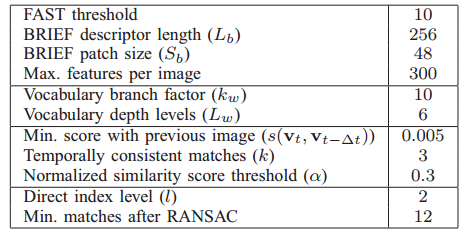
* 在建树过程中使用TF-IDF（term frequency-inverse document frequency）来表示特征的重要性: TF表示某特征在某张图像中频率，越大越好，IDF表示某特征在全部图像特征中频率，越小越好，所以用倒数
* 闭环检测的步骤如下，结合文后实验能更好理解各步的作用
* dataset query：通过查询KD树得到当前图片的M维表示向量，通过如下公式计算归一化的相似度分数，设定最小阈值得到候选的图片



* Match Grouping：由于时间上靠近的图片非常相似，可以将相似的图片分成不同组，得到不同组的相似分数H，公式如下，取最高分的一组用于下一步：



* Temporal Consistency（连续性检测）：计算之前query image是否与之前各组<vt−Δ t, VT1 >, . . . , <vt−kΔ t, VTk >，如果有k组通过测试，就取当前图片与T’组得分<vt,vt’>最高的作为候选帧进入下一步
* Efficient Geometrical Consistency：通过两帧之间的匹配点计算出基础矩阵F用于最后验证，保证至少有 12对匹配点。RANSAC方法也能验证局内外点。若匹配出错太多，结果肯定不好
* 实验过程中发现FAST+BRIEF匹配的特征都是中远距离的，中远距离在相机移动时尺度变化较小，间接说明了其尺度不变性较差
* 以下是实验的设定参数，在调参时可以作为参考



1. 《ORB-SLAM: Tracking and Mapping Recognizable Features》总结
2. 内容摘要

本文是ORB-SLAM的初步版本

1. 总计及理解

* 地图
* Map point or 3D ORB：一个map point和多个关键帧相连。nomal n向量：由map point到关键帧相机中心单位向量v的平均值；描述子取与其他所有帧描述子距离最小的描述子，加快直接将地图点与当前帧匹配的速度；dmax、dmin，根据scale和距离计算
* Keyframe：保存了提取的所有ORB特征和位姿；可以进行增加删除操作
* covisibility graph：结点是关键帧，边根据两个关键帧之间共视的路标点个数确定，一般阈值是15-30。并且边的权重就是共视点的个数
* place recognition
* recognition database：inverted index由特征找图片，在重定位和闭环检测任务中用于计算相似度。计算相似度时，首先收到一张新图片（query image）时，计算新图片与所有出现过这个特征的图片的相似度s。另外，按时间对关键帧分组计算相似分数和。
* efficient and refined ORB matching：分别计算相机位姿和相似变换，期间进行了几何验证。

direct index：由图片找对应结点和特征，在匹配任务中使用，使用暴力匹配需要n2次计算，从共同node计算可减少时间复杂度

* covisibility consistency：闭环检测的时间一致通过共视图判断，与DBOW2论文中有所区别
* tracking：通过两帧之间的具有3D坐标的匹配点使用3D-2D的EPnP算法+RANSAC计算位姿，再通过三角化计算一些其他匹配点的3D坐标
* ORB extraction：提取特征点
* Track from precious frame：跟踪未丢失时，特征匹配，利用3D-2D方法EPnP+RANSAC计算相机位姿
* Relocalisations：跟踪丢失时，计算当前帧的图片向量（bag of word），计算与关键帧的相似度，得到几张候选帧；进行特征匹配；利用3D-2D方法EPnP+RANSAC计算相机位姿。成功之后可以继续跟踪
* Track the local map：在计算出位姿后，利用投影找出更多匹配的路标点，最后使用double window相似方法计算最小重投影误差优化位姿。包括与当前帧有共视路标点的关键帧集合K1，及与K1各帧有共视路标点的集合K2，为了防止K2过大，设定了一个阈值。K1、K2集合构成了跟踪线程的局部地图。然后按论文上步骤建立其他路标点与当前帧之间联系（3D-2D）
* Final pose optimization：使用g2o+LM算法优化
* New keyframe decision：满足下面所有条件可插入
* 距离上次重定位超过20帧：重定位成功后帧会插入作为关键帧，然后继续跟踪；超过20帧保证重定位后得到的帧可以有效跟踪，在这20帧中跟踪不丢失
* 当前帧与参考帧之间至少有50个匹配点？还是路标点？：保证路标点足够多
* 当前帧的匹配点中路标点少于关键帧路标点的90%：保证相邻关键帧有一定不同，较容易满足
* 超过十帧且local map线程空闲。或者距离上一次插入关键帧超过20帧
* Local mapping：以插入帧为中心展开
* Bag of words conversion：得到插入帧的向量。计算插入帧在第二level的direct index
* Triangulate new map points：在covisibility graph中选择与插入帧共视点最多的N帧。将当前帧没有匹配到路标点的特征点使用描述子在level2与其他N-1帧没有匹配到路标点的特征进行匹配，去掉不满足极线约束的，之后进行三角化(2D-2D)标，丢弃视差较小的点。然后再按IV-D方法投影到其他
* Update the local area：在路标点被创建后的插入3中会检测路标点是否有效，需满足以下两个条件：
* 在期望看到它的帧数中有超过25%看到
* 创建后超过1帧（2帧）后，有至少三帧看到这个点

更新边和描述子D

* Local bundle adjustment：优化的范围在covisibility graph中与插入帧相连的所有帧，以及被这些帧看到的所有路标点。其他所有看到这些路标点的帧，不优化他们的位姿，用到他们的重投影优化路标点。优化方法LM+g2o+huber核。

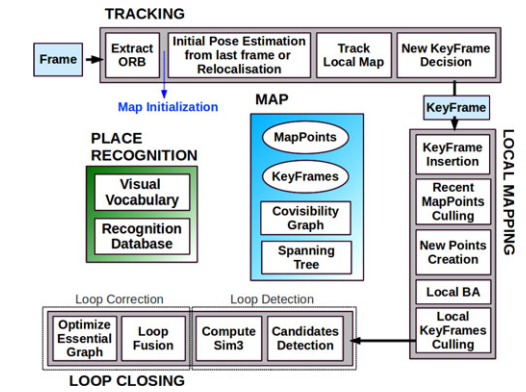
更新n、dmin、dmax

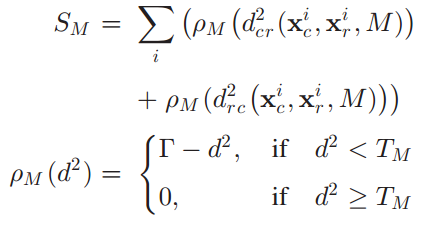
* Prune redundant keyframes：删除那些90%的点可以在其他3帧（同scale或者不同scale）看到的帧
* Loop closer：取local mapping线程处理的最后一个关键帧作为当前帧Ki
* loop detection：计算Ki与covisibility相邻帧最低的相似分数smin；保留covisibility中大于smin且不与当前帧相连的关键帧；然后做一致性检测，看是否有连续相邻3帧相似与当前帧。
* compute the similarity transformation：使用3D-3D（都是相机坐标系下）[8]+RANSAC直接计算Ki与闭环检测候选帧之间的相似变换，相似变换有7个自由度，s一开始为1，计算完后有可能改变，
* synchronization with local mapping thread：发送信号使局部地图停止工作
* loop fusion：使用相似变换融合闭环两头几帧，使其一致，以loop帧为准，使用Sil\*Tl计算当前帧位姿Ti。
* covisibility pose graph optimization：见文献[8]。因为BA有较严格的检测错误匹配的huber核，故要先做这一步
* global BA：优化方法LM+g2o+huber核。优化完后开启局部地图

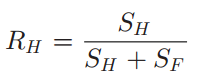
1. 1
2. 《ORB-SLAM: A Versatile and Accurate Monocular SLAM System》总结
3. 内容摘要

本文介绍使用ORB特征的单目SLAM算法：ORB-SLAM。包含了自动初始化、追踪、建图、重定位、回环检测等多个环节，可以在一个环境中长时间运行，取得了前所未有的表现。

1. 总结及理解

* parallel tracking and mapping (PTAM)的缺点：
* 缺少闭环检测，不能减小累积漂移误差和增强全局一致性
* 不能充分处理闭塞的情况
* 重定位时对视角变化的不变性低
* 地图初始化时需要人为干预
* ORB-SLAM有以下几个优点
* 整个系统使用同一种特征：ORB特征，使其更加高效、简单和可靠
* 可以在大的环境中实时运行
* 在essential graph上实现实时闭环检测，优化位姿
* 实时重定位，具有很好的视角和光照不变性
* 健壮的自动初始化
* 剔除冗余的帧和地图点，这也是使其能在环境中长期运行的原因
* 为了实现实时性，ORB-SLAM将整个程序分为tracking、local mapping、loop closing3个线程，整体框架如下图。文中第4-第7部分详细介绍了自动初始化、tracking、local mapping、loop closing的过程和细节，文献11中更详细介绍了重定位和闭环检测，需对照代码更进一步理解。
* 各线程的作用如下
* Tracking：根据摄像头传来的帧进行实时定位和决定何时插入关键帧。关键帧是摄像头所有帧中比较重要的一些帧，选择条件见V-E
* Local mapping：处理新的关键帧及优化局部地图来减小累积误差
* Loop closing：检测回环，进行full BA减小累积误差，建立全局一致地图
* 文中三幅图（graph）的关系：spanning treeessential graphcovisibility graph
* covisibility graph：结点-关键帧，边-共视地图点超过某个数（如15），个数即权重。用于局部地图优化，找出共视窗口
* essential graph：结点—所有关键帧，边—共视地图点超过某个更大的数（如30）。用于闭环检测pose优化
* spanning tree：结点—所有关键帧，边—共视地图点超过某个更大的数（如100）
* 自动初始化流程：
* 两帧匹配点大于某个阈值
* 分别使用DLT和八点法+RANSAC计算H和F矩阵，分别计算分数，分越高表现越好，迭代中取最高分的一组，若没发现合适的H、F返回步骤1



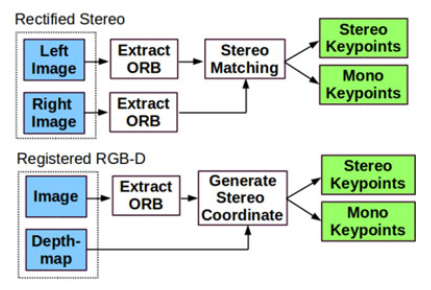
* 使用如下公式选择H或F
* 分别使用SVD等分解成位姿，得到8组和4组解，然后通过计算判断是否有一个解明显优于其他解，判断条件是具有最多在较大视差、相机前面和重投影误差较小的点。差别不明显，回到第一步
* Full BA优化
* Tracking：具体方法见上一篇总结
* 提取ORB特征，使用网格方法使分布均匀，每个cell中至少5个特征点，无论相机如何运动更容易跟踪。总数和每个cell数量都是自适应的
* 根据上一帧匹配的路标点3D-2D计算位姿初值，若匹配不够，则从上一阵周围帧匹配路标点，上一篇有方法
* 若跟踪丢失则进行全局重定位，从全局选出一些候选帧，通过这些候选帧的路标点进行重定位
* 跟踪局部地图中一些路标点，会检测局部地图所有的点；与上篇有些不同
* 选择是否插入关键帧，与上篇有些不同
* Local mapping：
* 插入关键帧：更新结点和边（与当前帧有共视路标点的），选择最大的边作为span tree的边。并计算整张图片的表示
* 删除一些路标点，在接下来三帧判断新建的路标点是否同，满足文中连个条件可认为通过，通过之后只判断第二个条件，这会在local BA丢弃异常点是出现
* 生成地图点，通过三角化，计算当前帧及与其相连的关键帧的匹配点（III-E）；丢弃不满足极线约束的点对，然后进行三角化计算；然后检查它们是否在两个camera下都为正深度，视差，重投影误差，尺寸一致性，满足则建立新的路标点；得到路标点后再将它们投影到与它们相连的帧，更新边权重
* Local BA，与上篇一样
* 删除关键帧，与上篇一样
* Loop closing
* 检测候选者：计算与相邻关键帧最低分数（在权重30边）smin；从inverter index库选择分数高于smin得帧；一致性检测，在covisibility中必须有三帧候选帧相连
* 计算当前帧和loop帧2帧之间相似变换
* Loop fusion，与上一篇一样，根据计算的相似变换更新闭环两头的数帧
* 对essential graph进行pose优化，实验证明进行该优化已经足够了，无需再进行full BA
* 闭环检测是在Essential graph上进行的，

1. 《ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo,and RGB-D Cameras》总结
2. 内容摘要

ORB-SLAM2是在ORB-SLAM的基础上进行改进，新增了将双目摄像头和RGB-D摄像头作为传感器时的预处理程序，同时实现了地图的重利用。

1. 总结及理解

* 下图为ORB-SLAM2的整体框架图，与ORB-SLAM相比，tracking和loop closing两部分不同。Tracking一开始需对三种不同相机做不同预处理，另外开启时会进行重定位重用建好的地图，识别自己的位置。Loop closing将sim3换成了se3，当进行full BA时，临时开启第四个线程
* 如下是双目和深度摄像头预处理过程，预处理后其他代码共用，使代码整个框架更清晰，便于移植，可读性高



* 双目和深度摄像头解决了单目尺度不确定性，可以从单帧图像获得深度，初始化更简单。也解决了单目旋转的问题
* 近特征点（小于基线的40倍）能很好的估计相机平移、旋转及特征点深度，远特征点只能很好的估计旋转，估计平移和特征点深度不准。这一点被用在了tracking插入关键帧的条件，有利于在视野开阔的室外场景下运行
* 程序中共进行了三次BA优化：tracking线程motion-only BA只优化位姿；local mapping线程local BA优化相机位姿和3D点；loop closing线程full BA优化闭环内所有位姿和3D点
* 下周阅读文献：
* Bundle adjustment a modern synthesis
* closed-form solution of absolute orientation using unit quaternions
* Double window optimisation for constant time visual SLAM
* Fast Relocalisation and Loop Closing in Keyframe-Based SLAM
* g2o: A General Framework for Graph Optimization