第17周汇报

——张溢炉

1. 《Motion and structure from motion in a piecewise planar environment》总结
2. 内容摘要

本文讲述了单应矩阵的求解方法

1. 总结及理解

* 当特征点位于同一个平面时，可以通过4对点求出相机位姿。
* 点共面满足如下条件

nTp1=d

其中n为平面的法向量（在第一帧相机坐标下），d为相机中心到平面的距离，p1为点在第一帧相机坐标系下的3D归一化坐标

* 由p2=Rp1+t1得p2=（R+tnT/d）p1，那么x2=K-1(R+tnT/d)Kx1，其中x为像素坐标，

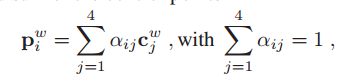
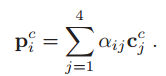
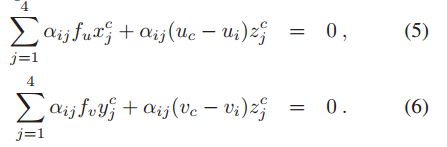
单应矩阵为：H=K-1(R+tnT /d)K，因为尺度不确定性，其具有一个自由度

* 由于尺度不确定性，可令：p2=（dR+t1nT）p1，令A= dR+tnT
* 通过8个点组成8个非齐次方程，将其转为超定方程，通过SVD求解超定方程求出A
* 通过SVD方法分解，当A的3个奇异值不同时得到8组n、R、t解，当3个中有2个相同时得到4组解。3个相同时得到一组未定义的n的解和一组有定义的解。

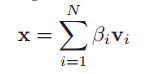
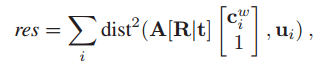
1. 《Accurate Non-Iterative O(n) Solution to the PnP Problem》总结
2. 内容摘要

本文讲述了一种由4组3D-2D点的不需要迭代的求解方法；EPnP，其时间复杂度为O（n）。

1. 总结及理解

* 3D（世界坐标P）-2D求解方法：首先，利用3D-2D点使用PnP方法求出2D点在当前帧坐标系的3D坐标p2（有尺度信息）；然后使用3D-3D的方法求出位姿，此时P=Twc2p2
* 很多PnP方法重在求解点在相机坐标系下的深度。而本文方法使用4个不在同一平面的control point（不是特征点的坐标）的加权和来求解。
* EPnP步骤
* 给出n个3D路标点，取四个控制点c，将n个点分别用控制点表示，由以下线性方程求出每个点的alpha，其中cj取个坐标的质心，其余三个控制点使用PCA求出，那么可得含4个未知数的4个方程组，可求出alpha
* 方程两边乘齐次变换矩阵T转到相机坐标系，方程仍然满足，则有
* 那么由相机的投影规则有，其中每个点的alpha值都可由第一步方程求出
* 可得如下方程，其中M为2n\*12，x为12维向量

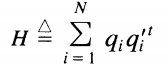
Mx=0

* 将M进行SVD分解，由最小的N个0奇异值（其不会严格为0，取最小的N个奇异值）对应右特征向量矩阵的向量求得x，理论上得6个点M刚满秩，由于尺度不确定性，会超定一个方程。实际中，常用15个点。但代码中使用ransac方法只用了4个点
* 当取N个向量时，求解方法不尽相同,都是利用相同点在不同坐标系的距离相等，具体看论文。N<=3时求解较简单，N>=3时需使用relinerization technique[16]。N越大，结果的精度会越高，但实际上，常将所有N（较小的N）的不同组合求出来，然后选取重投影误差最小的解。代码中先估值betas，在使用高斯-牛顿迭代求解
* 当所有点位于同一个平面时，只取3个control point即可，其他方法类推。

1. 《Least-Squares Fitting of Two 3-D Point Sets》总结
2. 内容摘要

本文讲述在已知3D-3D点的情况下求解R、t的最小二乘解方法，只需计算一次，用于orb-slam2重定位中

1. 总结及理解

* 步骤如下，具体证明看原文
* 计算各点的去重心坐标
* 计算H
* 使用SVD分解H，得到U、V
* 取R=X=VUt，计算det（R）=1,若为-1，第三行取反
* 再求t
* 1

1. 启发
2. 《Real-Time Loop Closure in 2D LIDAR SLAM》总结
3. 内容摘要

讲述谷歌cartographer算法

1. 总结及理解

* 匹配算法：
* scan-to-scan matching：会累计误差，最终误差会很大
* scan-to-map matching：减少了误差的累计（本篇论文使用方法）
* pixel-accurate scan matching：最后有说这个算法在后台用于将scan点集和最近的submap进行匹配，生成loop closing的约束条件。
* 优化算法
* 粒子滤波方法（particle filter）
* 基于图优化（graph-based）
* 1

1. 启发
2. 《A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization》总结
3. 内容摘要

本文介绍了double-layer ant colony algorithm（DL-ACO）算法，它使用了2层ACO，2层ACO单独、依次运行。首先，parallel elite ant colony optimization (PEACO)方法产生初始的无碰撞的路线，然后使用turning point optimization algorithm (TPOA)方法优化路径的长度、光滑性和安全性。最后，使用piecewise B-spline方法提高光滑性，更适合履带车通过。本文最后，通过实验证明方法的有效性和一致性。

1. 总结及理解

* 全局路径规划要解决的问题是：找到起点和终点之间无碰撞（动态和静态障碍物）和满足其他约束（距离、运动学等）的路径
* 评估路径的指标：长度、光滑性、安全性
* 本文主要贡献
* 增强函数（式1）和双层结构扩大了搜索范围，避免了过早收敛
* 路径优化方法优化了长度和光滑性
* piecewise B-spline在不影响安全性的情况下，优化路径的转角
* 本文方法相对其他方法能产生更短、更光滑的路径，此外，双层结构也增强了算法的稳定性
* 本文算法的缺点：本文算法需要精度高的地图，从而限制了它的应用场景。此外，本文算法只适用于静态场景。
* 传统的方法artificial potential field [4]和cell decomposition [5]被广泛应用，易于执行，但是计算量大，容易在复杂问题中掉入陷阱
* 与传统方法相比，启发式算法在路径规划中更有效。包括neural network [7], fuzzy logic

technique [8]，genetic algorithm (GA) [9], particle swarm optimization (PSO) [10]

ant colony algorithm (ACO) [11]

* PSO和GA方法由于初始值是随机选择的，相同情况下生成的结果具有不确定性。这会影响路径规划器的鲁棒性
* ACO被认为最有效、鲁棒性好的方法，在不同的场景中被频繁使用[12]-[15]。下面是ACO方法的拓展方法
* 文献[16]提出了三维的方法：unmanned combat aerial vehicle (UCAV)
* 为了在动态环境中使用，文献[17]在路径评估中增加了fuzzy cost function
* 文献[18]提出了基于动态路径重计算和改进的侦察（scout）蚁算法，可以完成机器人在未知环境的导航
* 但ACO存在收敛速度低、过早收敛的问题，此外，由于单步移动最大距离的受限造成的路线曲折，从而影响路线的长度和光滑性。下面是几种为了克服上述缺陷而设计的算法
* 文献[19]提出了hybridized ACO,它将局部搜索与已存在的ACO相结合使计算量减小
* 文献[20]提出了adaptive polymorphic ant colony algorithm（APACA），它采用了自适应的状态转移策略和自适应的信息更新策略
* 文献[21]提出了两层ACO，其将启发式算法分为了预处理层和路径规划层，这样可以避免局部最优。之后这种结构被广泛采用[22]-[25]，其可以扩大搜索范围、加快收敛速度
* 路径不平滑会加长运行时间，减少电机的寿命。平滑的路径能减少能力的消耗[26]，广泛应用的平滑方法有B-spline，如文献[31]、[32]。由于B-spline方法在某些场合会影响路径安全性，故本文提出piecewise B-spline的方法，它仅在拐角处作平滑处理。
* 算法步骤概述，详细公式见原文
* PEACO：产生初始路径

1. 计算增强因子：从式1可看出，迭代次数t越小，q越小
2. PEACO处理：
3. 初始化：将蚁群分为两个亚种群，初始它们参数，两个亚种群独立运行
4. 选择：没有到达终点的蚂蚁根据转移概率选择下一步
5. 信息素更新：当所有蚂蚁到达终点后，更新地图的信息素，以便下一次迭代使用
6. 达到最大迭代次数后，选择两个亚种群中最好的路径提供给下一层使用

* TPOA：通过移除不必要的转角等算法优化路径的长度、光滑性和安全性
* piece B-spline：使转角处变光滑
* 文献[35][36]介绍了运动控制规则
* 本文算法的运行时间和路径质量都与地图的分辨率正相关，在实际使用时要均衡两者

第16周汇报

——张溢炉

1. 《Fast Relocalisation and Loop Closing in Keyframe-Based SLAM》总结
2. 内容摘要

本文介绍了ORB-SLAM算法中重定位和闭环检测的方法

1. 总结及理解

* 提取500个角点时使用non-maximum suppression会改善角点分布，但是会降低匹配表现（特征点少了），故提取1000个角点
* 自动初始化时是2D-2D问题，重定位时3D-2D问题，闭环检测是3D-3D问题
* 在做一致性检测时，之所以选择3个方向，是因为在不同深度旋转的角度不一样
* 闭环检测：认为loop帧数据是比较准确的
* 选取候选帧
* 计算当前帧和闭环帧之间的相似变换Sk,l，算的过程中利用RANSAC作了几何检验。首先计算匹配点；利用无三维坐标匹配点同一张图周围最近3个有3D维坐标（两帧图都是用loop帧的路标点）的点计算该匹配点3D坐标（相机坐标系下），丢弃距离超过10个像素的点；对两帧进行上述操作后得到3D-3D点，然后利用文献[26]方法计算相似变换矩阵；然后验证，匹配点重投影小于6个像素表示是内点，如果在RANSAC前70次迭代中找到支持40%匹配点为内点的相似变换，则表示成功；最后利用所有内点在计算相似变换Sk,l
* 优化pose，因为BA有huber函数，漂移太大很难收敛。具体方法见文献[25]。detaSij是通过相似变换算出，闭环内两两帧之间变换，SjwSiw是由pose计算得出，r为residual error，理想状态等于1，这时两种方法恢复的值相同。目的是不断迭代优化pose，然后重复计算相似变换，使目标函数接近于0。协方差矩阵/\是指定的，不为单位矩阵时，表示不同自由度之间相互影响，但其常是单位矩阵。优化方法：g2o+LM+CHOLMOD



优化完pose后，利用如下公式校准每个路标点xj，使用Ti作为参考，接近于loop帧的位姿

* 路标点融合，如在loop看到的点，在当前帧又看到了，就会出现同一个物理点，在两帧中计算出了两个不同的3D坐标。将同一物理点（匹配点）在loop帧上世界坐标和当前帧该点的世界坐标一起投影到当前帧，若相距低于20像素，擦除当前帧的世界坐标。然后按同一的方法，融合loop周围关键帧与当前帧共视路标点，直到没有点可以擦除停止。在循环上诉操作，融合loop帧及其相邻帧与当前帧之前帧，直到没有点可以擦除。在点被擦除后就需要更新graph的边，包括帧之间以及帧与路标点的边
* 校正未参与闭环检测过程的新帧，在闭环检测当前帧之后前端又处理了一些新的帧。只需校正最新帧与闭环检测当前帧即可，其他帧会很快被局部优化。校正公式如下，这里只需利用尺寸校正的尺寸sk，l，注意校正前后标志

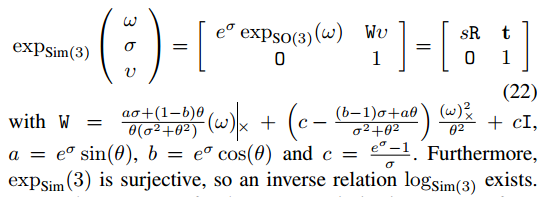


* 重定位：
* 为了更快的选出候选帧，不像闭环检测那样设置最小阈值alpha和一致性检测。在后面的几何检测以及在重定位之后跟踪超过20帧才会插入关键帧的设置，可以增强其鲁棒性。
* 使用闭环检测相似的方法恢复一些3D点，然后利用3D-2D+RANSAC计算位姿，如果在RANSAC前178次迭代中找到支持40%匹配点(至少20个点)为内点，则将该帧作为参考帧，若接下来超过20帧跟踪成功，重定位才算真正成功

1. 《Scale Drift-Aware Large Scale Monocular SLAM》总结
2. 内容摘要

本文主要讲述使用7dof优化的方法，没有说到两帧之间相似变换的求解方法，具体方法见下一篇。

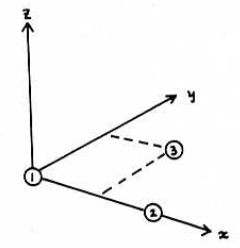
1. 总结及理解

* 单目相机有7个自由度会漂移，因为它是通过累加的方法计算尺寸，而双目、深度相机只有6个自由度，与单目相似，当前变换是不断利用上一次位姿获得的。这也是靠前的位姿和尺寸会更准确一些的原因
* sim（3）的李代数表示形式如下

1. 《Closed-form solution of absolute orientation using unit quaternions》总结
2. 内容摘要

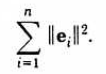
本文讲述一种计算已知一些点在自己坐标系下三维坐标的两帧图像之间的相似变换方法，该方法不用迭代，只需计算一次，使用于3个点及以上的情况。其中，将旋转用四元数表示。

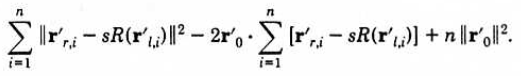
1. 总结及理解

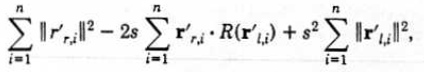
* 求解两坐标系之间相似变换方法分类：
* 单目具有7个自由度，那么取三个点就能找到9个约束，去掉其中两个约束便能求出解
* 迭代的方法：最小二乘方法
* close-form solution：如本文方法，给出匹配点在两个坐标系的坐标，其不需要迭代，只需一次计算，另外，其并不需要很好的初始值。
* 计算平移量：首先得到很多左右两个坐标系的匹配点。如下公式理解：假如r为向量，想象相机的带着一个坐标系在运动，先旋转，将两个坐标系方向变成一样，再变换尺寸，因为远近不一样，物理不变的长度，在近的坐标系变长，当然因为视角不同，物体形状还会畸变，这里先不考虑，故需要放缩一下；最后平移，r0是在右坐标系下的平移量。
* 具体方法
* 有选择的丢弃一些约束，使用已知在左右两个坐标系坐标的不在一条直线三个匹配点计算旋转。以其中2点作为x轴，2个点中一个作为原点，以3个点平面作为xy平面，将各轴化为单位向量。得到如下图坐标系，左右坐标系进行相同操作，这样就得到三个点在左右坐标系下的坐标系。因为原点相同，所以丢弃了平移，将各轴化为单位向量解决了尺寸问题，这样两坐标系只存在旋转，得到如下旋转公式，先转换为世界坐标系，再转换到右坐标系



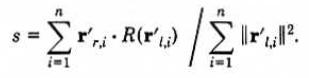
* 本文方法：给出n个匹配点在左右两个坐标系的3D坐标，最小化重投影误差，如下公式：



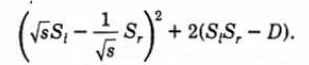
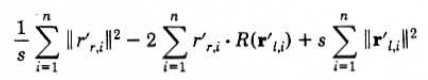
* 通过中心化可化为如下公式，第二项为0，第一项与s、R相关，第三项与r0相关
* 令第三项为0，有
* 第一项可化为不对称和对称两种情况：
* 不对称情况：两个坐标系测量精度差别很大时，将第一项进行恒等变换有：

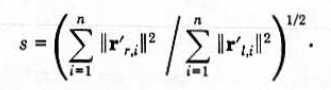


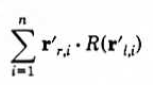
令第一项为0有：

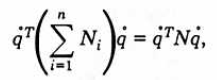
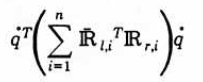


* 对称情况：两个坐标系测量精度差别不大时，将第一项进行恒等变换有



令第一项为0，这里s与R无关，有

* 对于以上两种情况，D越大越好，无论D多大约束的存在最后结果肯定大于0，D越大值越小，D公式如下
* 然后使用单位四元数可将其化为



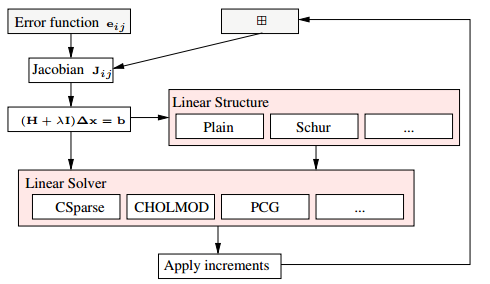
* A3证明欲求q使qTNq最大，只需求得N的最大特征值对应的特征向量作为q即可，从而求得旋转
* 然后使用上面s公式求出s
* 最后使用上面r0公式求出r0
* 当匹配点共面时存在两种特殊的旋转：旋转前后两坐标系交于一条直线和旋转轴为面的法向量。具体计算方法见论文

1. 《g2o: A General Framework for Graph Optimization》总结
2. 内容摘要

本文讲述g2o的原理

1. 总结及理解

* 整个框架如下图，g2o采用迭代方法操作，图中灰色部分error function和田操作（累加方式）是需外部指定的。另外，为了更高效，可以指定J；为了设计合适的求解方法，可以自己添加linear solver。在求解前可以先使用plain、schur方法消元。



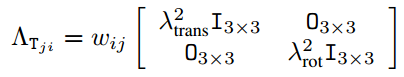
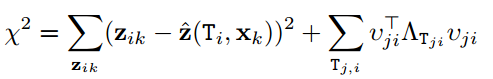
* 本文实验使用了CHOLMOD（cholesky）、PCG来求解。前者计算速度快，但不能利用参数的块结构。后者迭代次数多，计算速度慢，但能利用参数块结构。

1. 《Double Window Optimisation for Constant Time Visual SLAM》总结
2. 内容摘要

本文采用inner和outer窗口用于用于局部地图优化，利用了pose-point（Ti）和pose-pose（Ti,j）两种约束来优化。该方法能自动建立合适的graph，动态选择并优化inner和和outer窗口

1. 总结及理解

* Full BA太花时间，使用局部地图来优化能保持较好的实时性，包括以下方法
* Active windows：选取全部关键帧的一部分，通常窗口边界的关键帧是固定的，那些与它有共视点的帧会被包含。这种方法比较适合大范围、少闭环的场景，但不适合小范围、闭环比较多的情况，可能会因为包括的关键帧太多而很难收敛
* Relative representation：比较经典的是relative bundle adjustment（RBA），它优化的是两帧之间的相对位置。此方法适合少闭环情况，在闭环比较多的情况精度会下降。
* Pose-pose reduction：相对BA，其只优化pose，不优化路标点，路标点不变。
* Double window：以关键帧Vref作为参考帧，与其及其邻居相连权重最大的前M1帧被取为inner窗口W1，接下来M2帧被取为outer窗口W2，一般M1<<M2。W1帧使用BA优化，W2中一些帧也使用BA优化，W2的所有帧使用pose optimization。两者不需迭代进行，共同优化以下函数即可，优化完后以Vref为参考恢复计算了相对位姿的帧
* 闭环检测：用另一坐标系求出的路标点（两者的匹配点）投影到自己坐标系求出相似变换或齐次坐标，方便后面求残差



* 小闭环检测，首先文中定义了当前帧Vi的两级邻居N1、N2，小闭环候选帧是在N2中找到的，将Vi求出的的路标点投影到Vref求出位姿Tloop，然后删除一些重复的点。最后最小化下面目标函数优化



* 大闭环优化：和之前方法相似

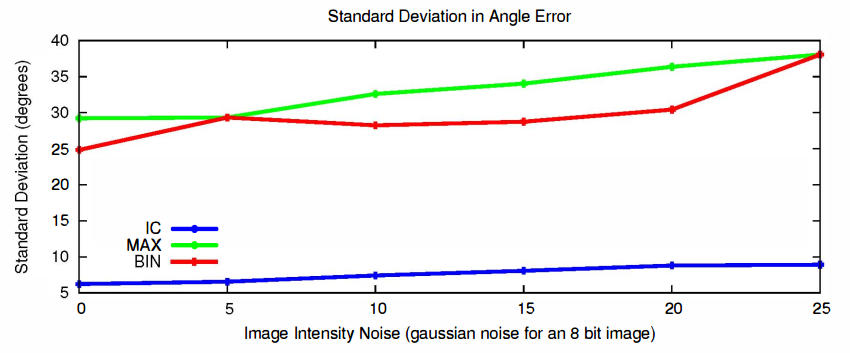
第15周汇报

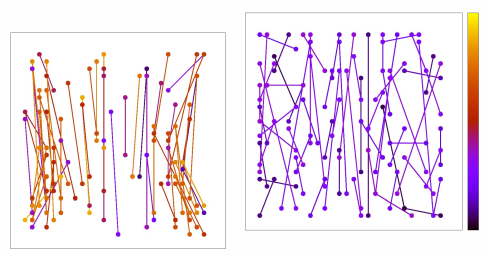
——张溢炉

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

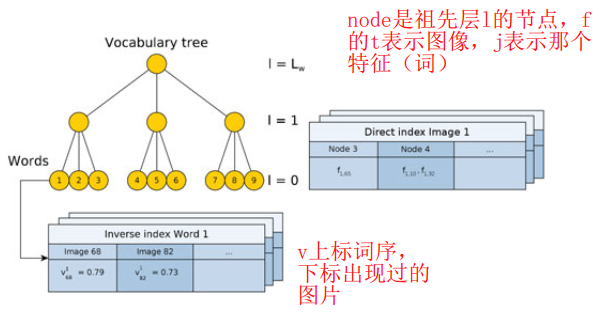
* 从实验结果来看，ORB特征具有较好的旋转不变性。但是使用图像金字塔的方法并没有使其具有较好的尺度不变性，还需改善

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

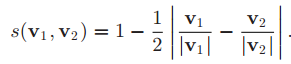
本文提出了一种从一串经过的图片中识别场景（Place Recognition）的新方法，其以FAST+BRIEF为特征，处理速度可达到22ms/帧，比之前的方法快了一个数量级，而且它能保证检测到的都是正确的

1. 总结及理解

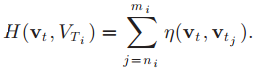
* Place Recognition对回环检测和重定位非常重要
* 存储图像描述子的树如下图，其中words由所有描述子的特征向量组成，direct index用于由图片查找其对应特征，inverse index用于由特征查找出现过的图片。先使用训练集图片获得足够大的树，然后再在运行更新direct index和inverse index。假如有M个特征（words），先使用K-means++将其聚成k类，然后再根据设定树的深度Lw，继续聚类。这样在查询特征时可以将时间复杂度变为log级。



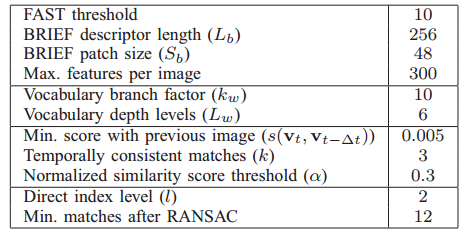
* 在建树过程中使用TF-IDF（term frequency-inverse document frequency）来表示特征的重要性: TF表示某特征在某张图像中频率，越大越好，IDF表示某特征在全部图像特征中频率，越小越好，所以用倒数
* 闭环检测的步骤如下，结合文后实验能更好理解各步的作用
* dataset query：通过查询KD树得到当前图片的M维表示向量，通过如下公式计算归一化的相似度分数，设定最小阈值得到候选的图片



* 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：



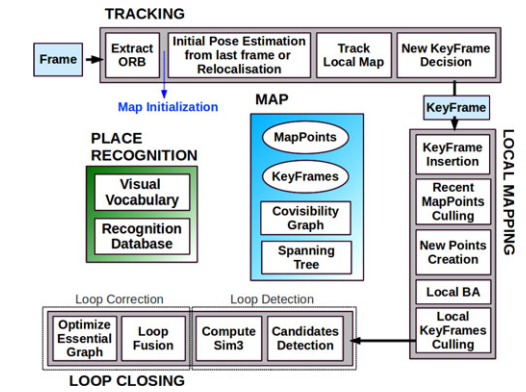
* Temporal Consistency（连续性检测）：计算之前query image是否与之前各组<vt−Δ t, VT1 >, . . . , <vt−kΔ t, VTk >，如果有k组通过测试，就取当前图片与T’组得分<vt,vt’>最高的作为候选帧进入下一步
* Efficient Geometrical Consistency：通过两帧之间的匹配点计算出基础矩阵F用于最后验证，保证至少有 12对匹配点
* 实验过程中发现FAST+BRIEF匹配的特征都是中远距离的，中远距离在相机移动时尺度变化较小，间接说明了其尺度不变性较差
* 以下是实验的设定参数，在调参时可以作为参考



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

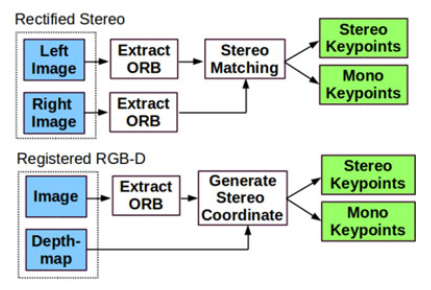
* parallel tracking and mapping (PTAM)的缺点：
* 缺少闭环检测，不能减小累积漂移误差和增强全局一致性
* 不能充分处理闭塞的情况
* 重定位时对视角变化的不变性低
* 地图初始化时需要人为干预
* ORB-SLAM有以下几个优点
* 整个系统使用同一种特征：ORB特征，使其更加高效、简单和可靠
* 可以在大的环境中实时运行
* 在essential graph上实现实时闭环检测，优化位姿
* 实时重定位，具有很好的视角和光照不变性
* 健壮的自动初始化
* 剔除冗余的帧和地图点，这也是使其能在环境中长期运行的原因
* 为了实现实时性，ORB-SLAM将整个程序分为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

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

第14周汇报

——张溢炉

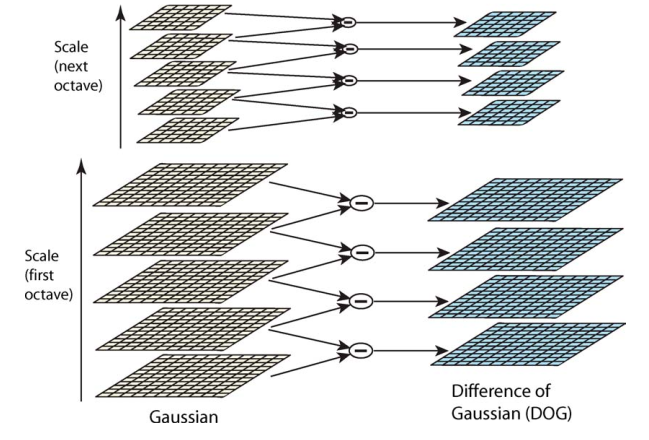
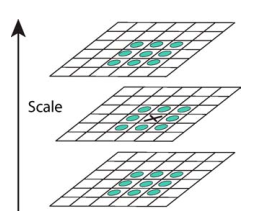
1. 《Distinctive Image Features from Scale-Invariant Keypoints》总结
2. 内容摘要

本文介绍了经典Scale Invariant Feature Transform (SIFT)特征。它具有尺度不变性（scale invariant）和旋转（rotation invariant）不变性，在仿射畸变（affine distortion）、 3D视点变化（change in 3D viewpoint）, 噪声增加（addition of noise）、 光照变化（change in illumination）的情况下匹配都具有较好的鲁棒性。另外，本文主要将SIFT特征用于物体识别，自己主要关注SIFT特征部分

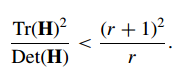
1. 总结及理解

* SIFT能从一张500\*500的图像中提取2000个左右特征，这对物体识别比较有帮助，在混乱背景下识别一个小物体至少需要3个特征才能比较可靠的识别
* Harris角点是经典的关键点，但是它对尺寸变化敏感
* SIFT特征提取主要分为四个步骤

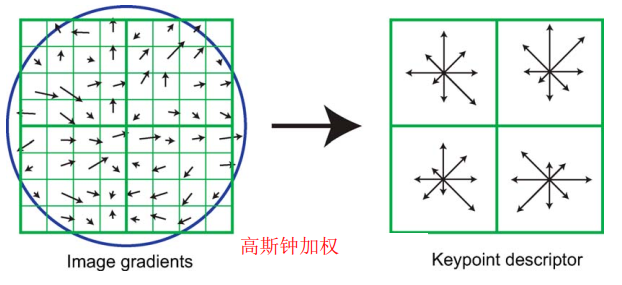
1. 尺度空间的极值点检测（Scale-space extrema detection）：得到的极值点为,包含位置和尺度信息。其中（x，y）是极值点的坐标，为极值点尺度，对应高斯核而得方差。计算步骤如下

* 利用高斯核、图像金字塔和difference-of-Gaussian（DOG）得到如下图所示高斯金字塔，左边为不同分辨率下通过不同方差高斯核得到的图像，右边为经DOG操作后的图像。具体计算公式见论文：如下图中，s=2，k=
* 取每组octave右边的中间两层来获得极值点，获取方法如下图，将带X的点与周围26个点比较，若该点大于或小于所有点，将其作为极值点

1. 关键点定位（Keypoint localization）：通过子像元插值和删除边缘效应对第一步得到的极值点进行筛选，最后获得关键点。子像元插值和删除边缘效应如下

* 子像元插值：利用泰勒公式得到连续情况下的D(xˆ)，当其低于某个值是将该特征点丢弃，文中在像素值归一化的情况下取0.03
* 删除边缘效应：为了得到稳定的特征点，只是删除DoG响应值低的点是不够的。由于DoG对图像中的边缘有比较强的响应值，而一旦特征点落在图像的边缘上，这些点是不稳定的。一方面图像边缘上的点是很难定位的，具有定位歧义性；另一方面这样的点很容易受到噪声的干扰而变得不稳定。一个平坦的DoG响应峰值往往在横跨边缘的地方有较大的主曲率，而在垂直边缘的方向有较小的主曲率。文中计算出hessian矩阵H后，通过如下公式筛选，取

1. 分配方向（Orientation assignment）：计算关键点局部梯度方向，将其作为特征点的方向
2. 计算描述子（Keypoint descriptor）：通过关键点周围区域梯度计算关键点描述子。如下图，将特征点周围8\*8的区域的梯度换算成每个方块包含8个方向矢量的2\*2描述子，其中包括2\*2\*8个描述子。文中取的是4\*4\*8个描述子。将描述子的计算与关键点方向结合起来便能达到旋转不变的目的



* DOG计算结果和使用拉普拉斯算子计算结果相似，即求出的是二阶导数
* SIFT特征的一个缺点是计算量太大，不适合用于对实时SLAM
* 待解决问题
* 将关键点与周围26点比较得到的极值点是角点和边缘点吗？其中数学含义是怎么样的？
* 描述子具体的计算过程是怎么的？

1. 《SURF: Speeded Up Robust Features》总结
2. 内容摘要

本文概述了具有尺度和旋转不变性的SURF (Speeded Up Robust Features)特征，相比SIFT具有更好的repeatability（可重复性，在不同视角的图像中被提取出来）, distinctiveness（区分度，区分不同特征点）,robustness（鲁棒性，在光照变化、视角变化、噪声等干扰下的可靠性高）。此外，它 计算量更小，实时性更高，这得益于其积分图像。

在一些不需要考虑旋转不变性的场合（如平面机器人导航，其只沿垂直轴旋转）可以使用upright SURF(U-SURF)，U-SURF不具备旋转不变性，不仅计算量更小，而且区分性更好

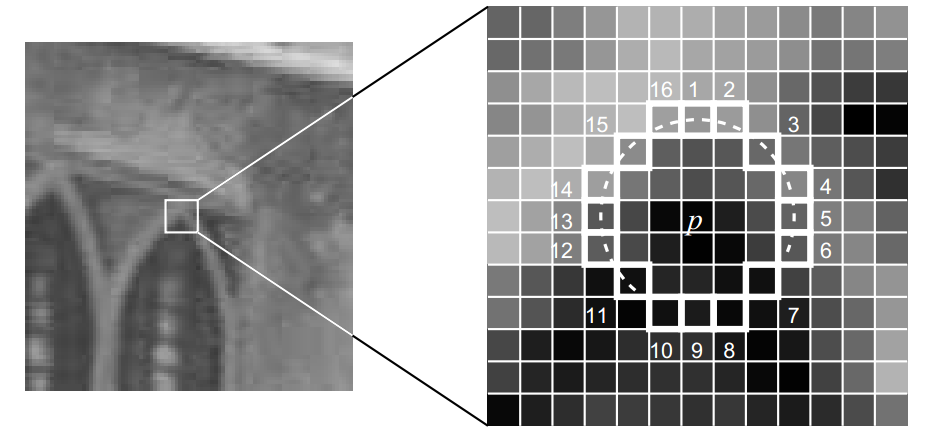
1. 总结及理解

* SIFT是经典的特征提取方法，但是计算量大，SURF是其改进型，计算量小一些，ORB在两者的基础上的改进型，计算更有效
* 特征提取和匹配的三个主要步骤
* 提取关键点，如角点、斑点和T形点。其需具备较好的repeatability
* 计算描述子，根据关键点周围的像素点计算其描述子，需具备较好的distinctiveness、robustness
* 匹配，根据描述子之间相似程度进行匹配，如计算两者表示向量的马氏距离、欧氏距离或汉明距离
* 本文第二部分对关键点和描述子的做了很好的回顾，有需要可以阅读其中相关论文加深了解
* SURF使用方形滤波器计算近似hessian矩阵来检测关键点，根据关键点周围Haar-wavelet响应的分布来计算描述子，两者都得益于积分图像来加速计算。描述子只使用了64D，这也加速了描述子计算和匹配时间。另外，作者还展现了一种新的索引方法，不仅加快了匹配速度，也增强了描述子的鲁棒性。特征提取也可分为以下四步
* 特征点检测：SURF使用Hessian矩阵来检测特征点，该矩阵是x,y方向的二阶导数矩阵，可测量一个函数的局部曲率，其行列式值代表像素点周围的变化量，特征点需取行列式值的极值点。用方型滤波器取代SIFT中的高斯滤波器，利用积分图（计算位于滤波器方型的四个角落值）大幅提高运算速度。
* 特征点定位：与SIFT类似，通过特征点邻近信息插补来定位特征点。
* 分配方向：通过计算特征点周围像素点x,y方向的哈尔小波变换，并将x,y方向的变换值在xy平面某一角度区间内相加组成一个向量，在所有的向量当中最长的(即x、y分量最大的)即为此特征点的方向。
* 特征描述子：选定了特征点的方向后，其周围像素点需要以此方向为基准来建立描述子。此时以5\*5个像素点为一个子区域，取特征点周围20\*20个像素点的范围共16个子区域，计算子区域内的x、y方向(此时以平行特征点方向为x、垂直特征点方向为y)的哈尔小波转换总和Σdx、ΣdyΣdx、Σdy与其向量长度总和Σ|dx|、Σ|dy|Σ|dx|、Σ|dy|共四个量值，共可产生一个64维的描述子。
* 待解决问题
* 理解hessian矩阵和特征点之间联系
* 理解哈尔变换

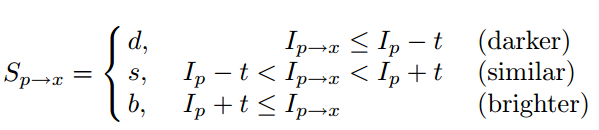
1. 《Machine learning for high-speed corner detection》总结
2. 内容摘要

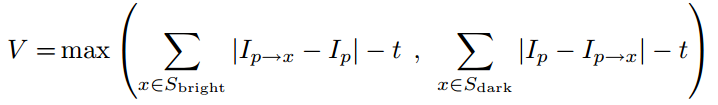
本文介绍了一种使用机器学习提取FAST（Features from Accelerated Segment Test）角点的方法

1. 总结及理解

* 本文第一部分系统介绍角点检测的发展，有需要可以阅读其中相关论文加深了解
* 快速提取角点的方法FAST：首先取如下图所示中心点p周围的16个点，若有连续n个点像素值大于或小于点p，且都超过阈值t，则认为p为角点。当取n=12时，称其为FAST-12。相对于n<12的情况，FAST-12有一种快速判断不是角点的方法，先判断1、9点中是否有一个大于或小于p且超过阈值t，若没有则p点不是角点；若有，则继续计算5、13，判断四个点（1，5，9，12）中是否有三个点满足阈值要求，若不满足，则不是角点；若满足。则继续判断是否有连续12个点满足条件。
* FAST-12存在下面四个问题

1. 当n<12时获得的候选点比较多
2. 检测出来的角点不是最优的，因为它的效果取决于要解决的问题和角点的分布情况
3. 对于角点分析的结果被丢弃了
4. 检测到的很多角点都是连在一起的

* 本文提出利用机器学习的方法解决FAST-12的前三个问题，但机器学习有一个问题，训练集不可能包含所有样本，故预测时需借助模型的泛化能力，那么结果有一定的置信度
* 首先根据以下公式将每个中心点p的周围16个点分成3类，组成一个关于中心点p的16维的特征向量，并将每个中心点用布尔型Kp标注为是否是角点
* 然后使用这些数据训练ID3决策树
* 最后使用训练好的决策树来预测角点
* 另外，本文提出使用极大值抑制（Non-Maximal Suppression，NMS）的方法解决FAST-12的第4个问题。方法如下

1. 对所有检测到的角点构建一个打分函数（如下式）。就是像素点与周围16个像素点差值的绝对值之和
2. 考虑两个相邻的角点，并比较它们的分值。
3. 分值较低的角点将会被删除。

* 导数值易受噪声干扰
* 简单介绍了评价指标consistency of corner numbers(CCN)、ACU
* 使用机器学习方法提取FAST角点的方法具有以下优缺点
* 优点

1. 提出速度快
2. 对于变化大的图片和不同种类特征具有较高的可重复性

* 缺点

1. 抗干扰能力不太好
2. 可能对一个像素宽的直线作出响应
3. 依赖于阈值的选择

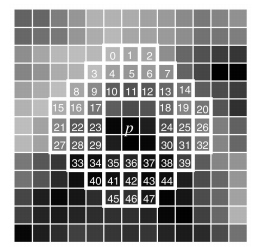
* 待解决问题
* 阅读文中相关论文理解角点与H矩阵及其特征值之间的联系？
* ID3的训练方法？

1. 《Faster and Better: A Machine Learning Approach to Corner Detection》总结
2. 内容摘要

本文是上一篇文章方法的改进版，它更快、更好

1. 总结及理解

* 较上一篇本文对提取角点的方法做了更好更全的分类，可以选择阅读
* 本文提出了FAST-ER（Enhanced Repeatability），它在原来算法里提高特征点检测的重复度，重复意味着第一张图片内的检测的点，也可以在第二张图片上检测出来。由于一些形变较大因素造成的形变，很难通过简单且固定的模板将所有的角点检测出来，而原来的FAST算法其决策树并不能最优的实现区分角点（实现最优的重复率）。FAST-ER就是针对这样的问题而提出的，其主要是通过模拟退火（也有通过最速下降法的）优化原先决策树的结构，从而提高重复率。
* FAST-ER选择了如下图所示中心点周围48个像素点来判断是否是角点

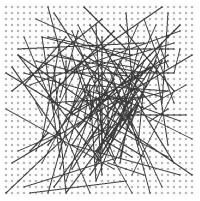


* FAST-ER的作用和总结：因为每个迭代过程中，都需要对重新应用新的决策树进行检测，而且16个变换中每一个都需要对应一个候选树，所以这样的检测算法并不十分有效，因此，从效率上考虑，上述的算法一般用于产生训练数据，之后获得较为精确的角点检测结果，我们就可以通过原来的FAST算法来产生单个树
* 待解决问题
* FAST-ER使用模拟退火方法训练的具体过程

1. 《BRIEF: Binary Robust Independent Elementary Features》总结
2. 内容摘要

本文提出了BRIEF（Binary Robust Independent Elementary Features）描述子，它是一种使用二进制串表示的描述子，使用汉明距离做匹配，占用存储更小，计算速度更快

1. 总结及理解

* BRIEF使用对关键点周围一块S\*S区域的（x，y）像素点对做比较来获得它的描述子，如下图所示，实线表示一对比较的点，其使用(X, Y) ∼ i.i.d. Gaussian(0, S2/25)分布获得。提取BRIEF描述子包括以下几步
* 使用高斯核对图像进行平滑处理
* 使用某种分布选择关键点周围的点对，常用如上图所示高斯分布。然后比较点对之间的大小获得关键点的描述子，点对和比较顺序选定之后不再改变
* BRIEF-k表示有k\*8对点作比较

1. 启发

要保持特征点的尺度不变性，常使用图像金字塔，不同距离拍摄的照片形成图像金字塔一样的效应。旋转不变性一般通过计算关键点周围像素特点（如梯度）确定关键点的主方向，这里旋转一般指摄像头绕与照片平面垂直轴旋转，这样关键点的主方向能反映拍摄时旋转的角度，然后将这个角度作为描述子的考虑因素提高匹配的鲁棒性

第13周汇报

——张溢炉

1. 《Mapping Technology in Visual SLAM: A Review》总结
2. 内容摘要

本文主要介绍一些传统的建图方法

1. 总结及理解

* 对于移动机器人来说，SLAM作为底层技术，可以用于上层定位、导航、避障、重建和可视化等，稀疏地图可以用于机器人定位，但导航、避障和重建需要用到稠密地图。
* 视觉SLAM可以用于建立稀疏、半稠密和稠密地图。
* 基于特征的方法主要用于建立稀疏地图，Mur-Artal [9]提出了一种建立半稠密地图的方法。直接法可以用于建立半稠密和稠密地图，对环境特征和图像质量不太敏感。稠密地图可以用建立网格地图和八叉树地图，用于导航和避障
* ORB-SLAM的各个优点，它可以用于室内和室外环境：

1）新三维点的匹配更高效

2)闭环检测可以解决长期漂移积累误差

3)目标函数相位可以非常有效地求解

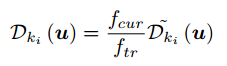
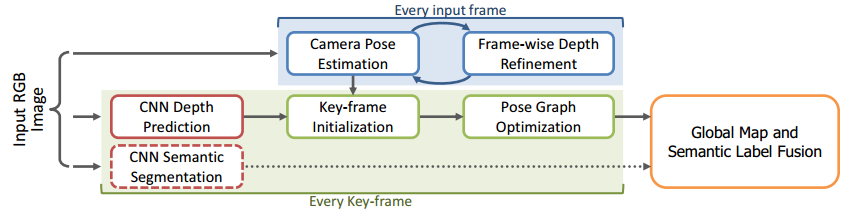
* 建立稀疏地图计算量较小，可以使用板上的处理器。但建立半稠密和稠密地图计算量较大，通常需要用GPU
* 未来的发展方向和应用有3D实时建图、建立网格地图和拓扑地图、增强现实AR、医疗应用

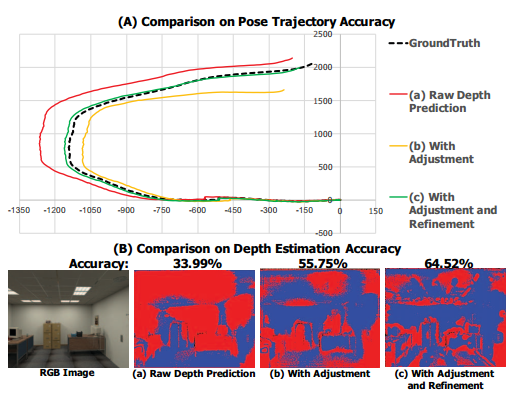
1. 《CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction》总结
2. 内容摘要

本文介绍了将CNN与半稠密SLAM方法结合的单目相机稠密建图方法，另外还在图中加入了语义信息

1. 总结及理解

* 动机：克服单目相机的一些局限，如绝对尺寸不确定性、旋转和低纹理区域遇到的问题
* 模型如下图所示，模型中有三个通道，中间的是CNN通道，可以通过预训练好的CNN模型生成全局的初始化深度，CNN模型建立的深度在物体边缘比较模糊；这正好与半稠密方法互补，故再利用最上层的半稠密方法来精细化物体的边缘；最下层通道与上两层相互独立，增加语义信息有助于智能体对周围环境的理解和相互作用。注意，因为实验使用的摄像头与采集数据集使用的摄像头不同，其内参也会不一样，所以使用CNN预测的深度需要经过如下公式调整



* 下图展现的是调整CNN预测参数（with adjustment）和精细化（Refinement）带来的效果提升
* 文中使用的两个数据集是：the ICL-NUIM dataset [8] and TUM RGB-D SLAMdataset [26]
* 语义标记中常用的四个标记是：floor（如地板）,vertical structure（如墙）, large structure（如家具）, small structure（如杯子）
* 实验时，CNN和语义分割模型是在GPU运行的，其他运算在CPU上运行。目前来看深度学习的计算量太大，需要使用到CPU和GPU，不太适合计算能力弱、实时性要求高的小型移动机器人，但5G等通讯技术在速度上的提升，可以考虑将数据传输到云端处理。此外，像汽车这种大型的移动机器人，可以搭载GPU使用深度学习来做SLAM，有利于提高自动驾驶的安全性
* 未解决问题：具体公式还需在理解一些基础公式（如g2o、Huber function等）后来理解

1. 《Past, Present, and Future of Simultaneous Localization and Mapping: Toward the

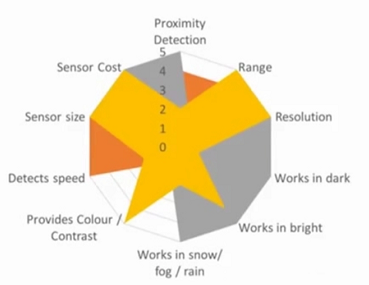
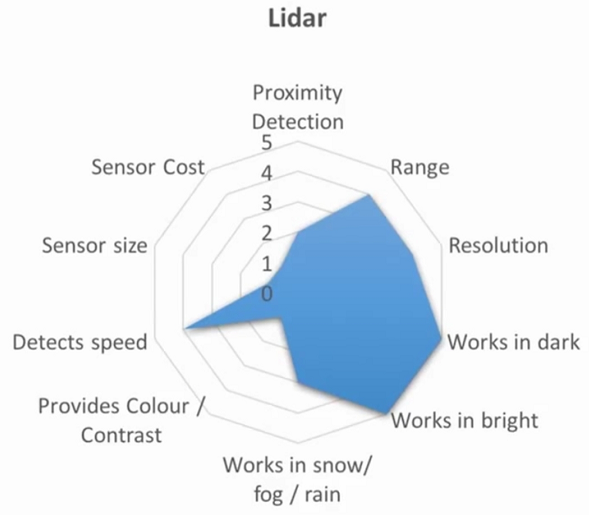
Robust-Perception Age》总结

1. 内容摘要

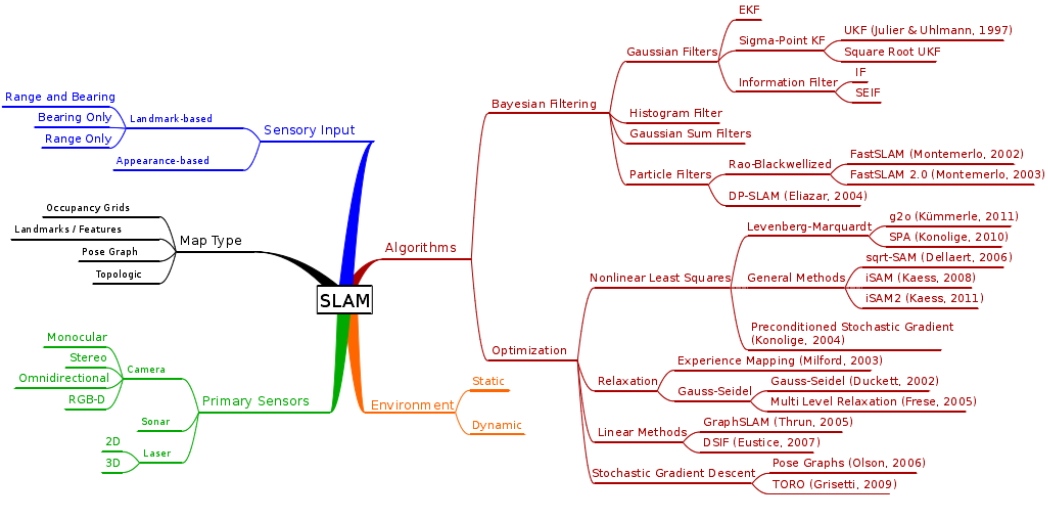
本文较上周两篇对SLAM过去30年的发展做了比较完整的回顾，不仅仅是视觉SLAM。可将SLAM的发展分为2个时代：古典时代（1986-2004）、算法分析时代（2004-2015）

1. 总结及理解

* 建图主要有两个作用：第一，给上层算法使用；第二，可以减小视觉里程计估计的误差
* 以下分别是超声波、摄像头、雷达、激光的优缺点图，单个传感器有一些局限，融合多种传感器可以取长补短，适应不同环境



* visual-inertial navigation (VIN)方法融合了相机和IMU，不用做闭环检测，取长补短。相机适合长时间移动测位姿，但相机不能判断是物体动还是相机动。IMU擅长快速移动情况，可以用于判断相机的短时间的移动
* 状态估计的两种方法
* 历史上很长一段时间，研究者们使用滤波器，尤其是扩展卡尔曼滤波器（EKF），只关心当前状态
* 近年来普遍使用非线性优化方法，使用所有时刻采集的数据进行状态估计，被认为优于滤波器，而成为主流方法
* SLAM的常见公式有：maximum a posteriori estimation, factor graph optimization, graph-SLAM, full smoothing, smoothing and mapping(SAM)
* SLAM的分类如下图，从算法、环境、传感器、地图类型、传感器的输入进行了分类

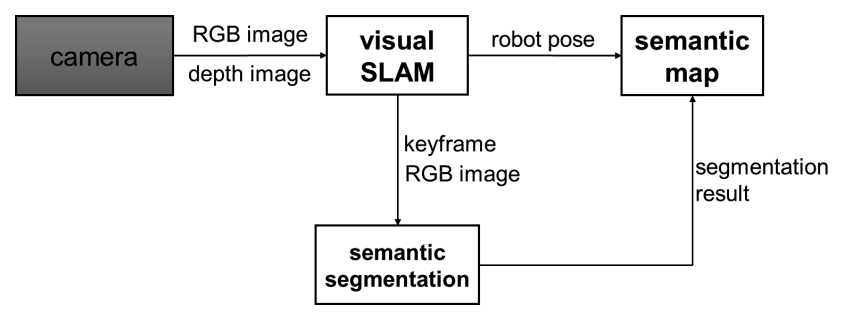


1. 《Real-Time Semantic Mapping of Visual SLAM Based on DCNN》总结
2. 内容摘要

本文提出了一种使用DCNN模型建立语义地图，并具有较好的实时性

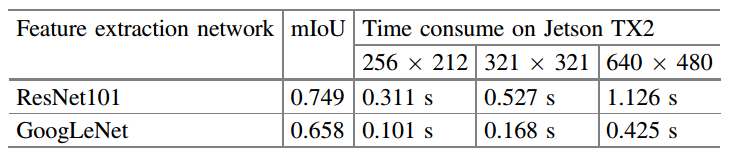
1. 总结及理解

* 动机：传统的SLAM只包含距离和颜色等低级信息，本文使用深度学习在地图中加入语义高级信息，并保持SLAM的实时性，促进机器人对陌生环境的理解，加强人机交互
* 本文的整体模型如下，首先使用ORB-SLAM选择关键帧，实时定位和建图，然后将关键帧通过语义分割模型进行处理，得到语义图，最后将ORB-SLAM建立的3D图与语义图合成。



* 语义分割模型如下，包括feature extraction、muti-scale、classification三个模块
* feature extraction：使用了一个1\*1的卷积层将64个通道减少到32个通道加速后面运算；使用dilated convolution卷积提高输出特征的分辨率
* muti-scale：先使用图像金字塔方法池化，再使用4个dilation为6, 12, 18，24及核为3\*3的dilated convolution并行处理，最后将结果串联得到最后特征
* classification：使用softmax进行分类



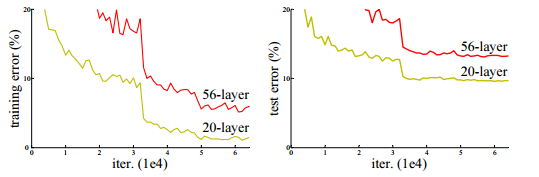
* 本文使用了NVIDIA Jetson TX2嵌入式模块来实现视觉SLAM的实时处理。它是一台模块化 AI 超级计算机，采用 NVIDIA Pascal™ 架构，具有 256 个CUDA核心。更棒的是，它性能强大，但外形小巧，节能高效，非常适合机器人、无人机、智能摄像机和便携医疗设备等智能边缘设备。它支持 Jetson TX1 模块的所有功能，同时可以铸就更大型、更复杂的深度神经网络。
* 实验结果如下图所示，虽然准确率相对ResNet101低0.091，但在各种分辨率下速度是ResNet101的3倍左右。从本文实验结果来看，将深度学习用于小型移动机器人具有一定可行性

1. 《Deep Residual Learning for Image Recognition》总结
2. 内容摘要

本文是ILSVRC 2015分类任务的冠军模型：ResNet

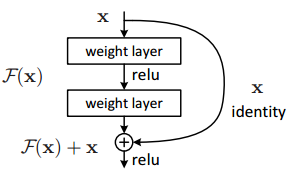
1. 总结及理解

* 动机：一般来说，随着模型深度增加，在不过拟合的情况下，训练误差和测试误差会更小，但是实验结果并不是这样，刚刚相反，出现了退化现象，实验结果如下图所示。ResNet旨在解决这个问题。假设：如果最优函数更接近于恒等映射而不是零映射，那么solvers应该更容易找到与恒等映射相关的扰动，而不是将函数作为一个新的函数来学习，这样学习更快。

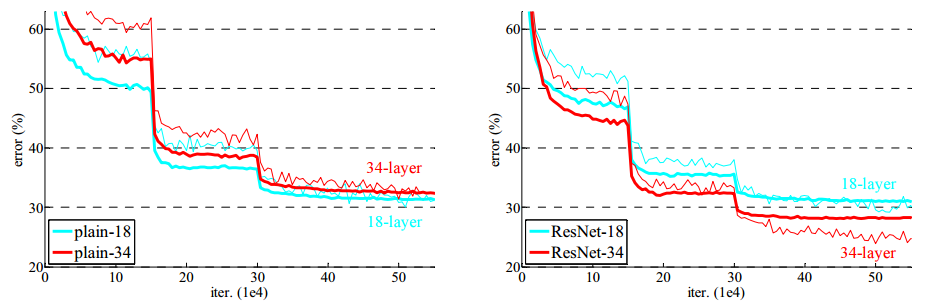


* ResNet由堆叠的一个个小模块（如下图所示）组成，相对VGG的改变，ResNet的非线性层拟合的目标函数是F(x) := H(x) – x，H(x) := F(x) – x是任务待拟合的函数。其中X的映射有三种方式，从实验结果来看A\B\C对实验结果影响不大，但计算量逐个增大：

1. 当非线性层输出维度和x相同时，对应元素直接相加；不同的话，填充0
2. 当非线性层输出维度和x相同时，对应元素直接相加；不同的话，x进行线性变换，然后与F(x)对应元素相加
3. 任何情况下，x都进行线性变换，然后与F(x)对应元素相加



* 下图为plain网络和ResNet实验结果，细线为训练误差，粗线为测试误差。ResNet-34的训练和测试误差都小于ResNet-18，说明ResNet解决了退化的问题



* 待解决问题：batch normalization (BN) [16]

第12周汇报

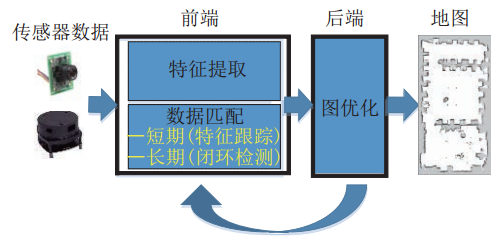
——张溢炉

1. 《基于深度学习的视觉 SLAM 综述》总结
2. 内容摘要

本文介绍了同时定位与地图构建（SLAM），介绍了一些传统的方法和深度学习方法

1. 总结及理解

* 典型SLAM系统前后端如下图，传统的方法包括运动方程和观测方程，运动方程利用两帧图像计算相机位姿，观测方程用于后端优化。单目摄像头较难计算像素点的深度，需要通过两帧图像计算；双目可以通过一帧图像计算深度；RGB-D摄像头直接得到深度
* 帧间估计：也称视觉里程计，通过前后两帧图像计算摄像头位姿
* 后端优化：优化相机运动轨迹，减小噪声的干扰和累积误差
* 闭环检测：判断自身是否进入历史同一地点．闭环检测发生时可触发 SLAM 后端全局一致性算法进行地图优化，消除累积轨迹误差和地图误差．闭环检测问题本质上是场景识别问题
* 地图构建：利用传感器得到的数据建立三维地图

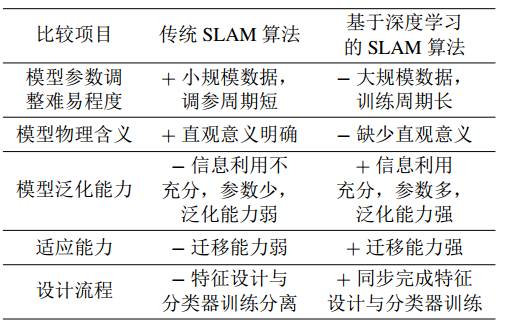


* 人工设计的稀疏图像特征当前有很多局限性，一方面如何设计稀疏图像特征最优地表示图像信息依然是计算机视觉领域未解决的重要问题，另一方面稀疏图像特征在应对光照变化、动态目标运动、摄像机参数改变以及缺少纹理或纹理单一的环境等方面依然有较多挑战
* 目前深度学习结合SLAM有3个方向
* 用深度学习方法替换传统SLAM一个或几个模块特征提取，如特征匹配，提高特征点稳定性，提取点线面等不同层级的特征点，深度估计，位姿估计，重定位；
* 在传统SLAM之上加入语义信息，如图像语义分割，语义地图构建；
* 端到端的SLAM，端到端是输入image输出action，没有定位和建图。 如机器人自主导航（深度强化学习）等。

本文介绍了基于深度学习的帧间估计、闭环检测和语义地图生成，目前深度学习多应用于 SLAM 局部的子模块，如定位模块或闭环检测模块

* SLAM如果能和一些CV、nlp任务结合起来，会产生一些比较好玩的任务，比如：
* 建图时利用图像描述生成一段自然语言描述，并保存当前帧，直到下一帧与当前帧的相似度低于某个阈值时，再生成一段描述，当语句到达一定数量或者描述完一个场景，再利用nlp中生成结尾的任务生成一段总的描述
* 建图时识别图中物体，然后从外部知识获取物体的属性（如颜色、是否可移动等）、物体间的关系，建立某空间的语义图谱
* 与EQA任务结合，SLAM可以实时建图，可以更新某个空间被移动过的物体，但是实时建的图会有很多噪声，不像仿真环境中那么干净

以上任务在小范围比较合适，如家里、仓库等

* 传统方法与深度学习方法比较如下图

1. 《Ongoing Evolution of Visual SLAM from Geometry to Deep Learning:Challenges and Opportunities》总结
2. 内容摘要

本文介绍了视觉SLAM的一些传统的方法和与深度学习结合的方法，讲述了深度学习应用到视觉SLAM上面临的一些机遇和挑战

1. 总结及理解

* 传统的方法如下：
* 基于特征（Feature-Based）：首先通过特征匹配找出两帧图像相同的像素点，可以通过关键帧的位姿计算这些像素点的世界坐标，然后利用几何算法计算相机位姿。因为相机可以看成一个小孔成像的模型，对于单目相机具有尺度不确定性，需要通过两帧图像来计算像素点的深度。整个计算过程中默认前一帧图像的结果是对的，所以会有累积误差，这就需要利用后端和闭环检测来减小累积误差。后端采用图优化的方法，将一些路标点投影到相机成像画面，通过一些算法（如BA）使投影点和实际点尽可能接近。比较有代表性的特征有SIFT、SURF和ORB，目前ORB用的比较多。得到位姿和深度后，建图还是比较容易，单目使用这种方法只能建立稀疏的三维图
* 直接法：也称光流法，有稠密和半稠密的方法。假设相邻两帧图像的像素值不变，通过梯度下降的方法调整位姿，使两幅图像对应像素点的方差最小，这就要用到李代数，因为旋转公式有约束条件，不好做梯度下降。可以进行稠密和半稠密建图。目前比较好的方法是Direct Sparse Odometry (DSO)
* CNN、RNN、LSTM、encoder-decoder、无监督学习在视觉SLAM方面都有应用，引入了CV和NLP方面的大多数模型。文中主要提到深度学习在SLAM的深度估计、位姿估计、语义建图（如下图）三个方面结合比较多。
* 数据集
* 深度估计：KITTI [35], TUM [40], NYU [41]
* 重定位：7-scenes dataset [46]，Cambridge landmarks [33]
* 视觉里程计（VO）：KITTI [35], Robotcar [38],M’alaga [47], EuRoc MAV [39], NYU [41],TUM [40]
* 场景分割：PASCAL VOC [42], NYU [41], Synthia [43],Cityscapes [37], KITTI [35],ADK20 [45]
* 传统的方法在定位和建图已经具有比较好的准确性和实时性，但鲁棒性不太好，会面临特征稀少、光强变化、尺度不确定性、相机内参漂移等问题，将深度学习应用到SLAM可提高其鲁棒性、语义信息以及学习能力，可以使robots更好地与环境交互。CNN可以提取大量的隐性图像特征，具有尺度不变性、旋转不变性。
* 监督学习需要数据集来驱动，但不同robots面临不一样的未知环境，很难建立通用的数据集，建立数据集也需要花费大量的人力和时间，这是当前面临的瓶颈。无监督学习、传统方法与深度学习的结合、多传感器（雷达、超声波、激光、陀螺仪）的融合有利于解决目前面临的一些问题。

1. 《Visual SLAM for Automated Driving:Exploring the Applications of Deep Learning》总结
2. 内容摘要

本文介绍了视觉SLAM在自动驾驶上的应用，很多内容和之前两篇相似，在这就不再重复

1. 总结及理解

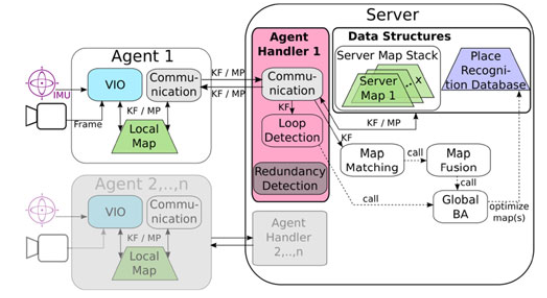
* 视觉SLAM比较适合自动驾驶的以下三个情景
* 停车：停车时要识别周围物体，选择车位，避开障碍物。在外面停车每次都得重新建图，对于生活中常去的停车的地方，如家里，可以将图存储下来，方便下次使用，虽然每次物体放置都不一样，但停车场整体结构一般不会变，有助于重定位
* 高速路：高速路需要比较高的实时性，每秒需要处理30张图片，传统方法实时性较好
* 城市驾驶：城市环境比较复杂，移动物体较多，DSO-SLAM表现较好
* 自动驾驶对安全性要求很高，因为一个失误可能就是人命，似乎很难找到一个置信度百分之百的深度学习模型，需要综合各种方法来防止意外的发生。因为世界各地环境都不相同，也很难找到一个通用的数据集
* 文后还提到了一些CNN Based Pipelines，如Tateno et al. [54]提出联合学习语义分割和建立深度地图

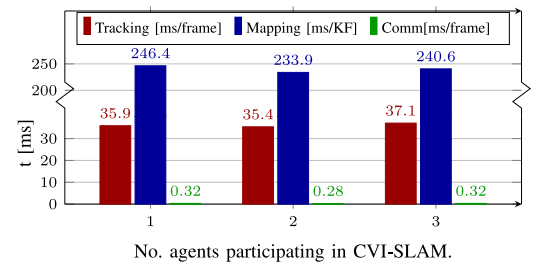
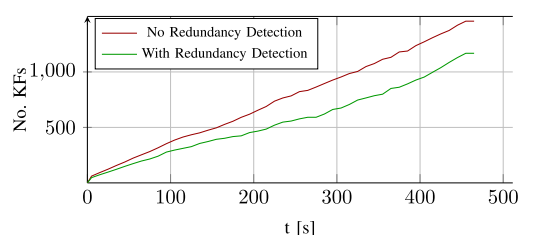
1. 《CVI-SLAM—Collaborative Visual-Inertial SLAM》总结
2. 内容摘要

本文介绍了多个智能体协作的Visual-Inertial SLAM，每个智能体上搭载了摄像头和Inertial Measurement Unit（IMU），与中央处理器协同定位和建图

1. 总结及理解

* 整体架构如下图所示，agent负责处理实时性较高的计算，如定位。Server负责处理计算量大且实时性不高的计算，如冗余检测、闭环检测、地图融合、BA。这样可以利用server强大计算力，也可以保持agent相对独立，但是这对两者之间的通信速度和质量要求比较高，图像的数据量比较大。从本文实验结果来看，该方法在agent数量上具有较好的可拓展性，对网速的需求可以接受。



* 下图为不同agent在frame tracking, local mapping of the VIO, communication所用时间，通信时间相对处理时间还是比较少的，但随着agent的增加，agent的处理量不变，但通信压力会变大，server的处理数据量会线性增大
* 下图是在4个agent情况下有无冗余检测模块随着时间增长待处理关键帧的数量变化图，由图可见冗余检测可以有效减少待处理关键帧的数量，减少server的处理压力
* 从本文大致了解了CVI-SLAM的方法，里面涉及的很多公式还需以后继续弄明白