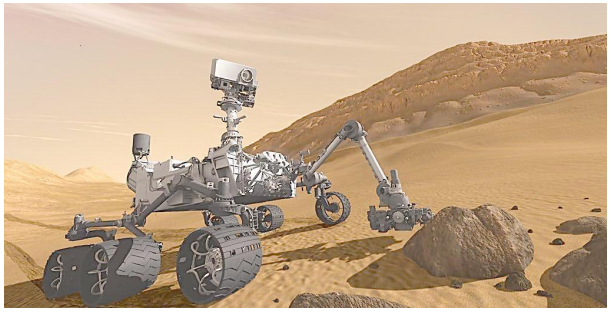
视觉SLAM

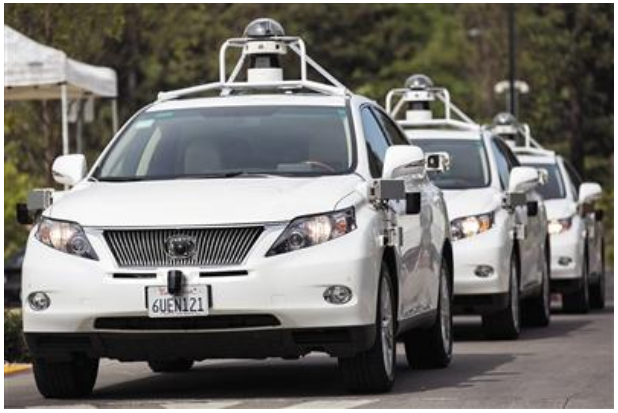
章国锋

浙江大学CAD&CG国家重点实验室

SLAM: 同时定位与地图构建

- 机器人和计算机视觉领域的基本问题
  - 在未知环境中定位自身方位并同时构建环境三维地图
- 广泛的应用
  - 增强现实、虚拟现实
  - 机器人、无人驾驶、航空航天





SLAM常用的传感器

- 红外传感器：较近距离感应，常用于扫地机器人。
- 激光雷达、深度传感器。
- 摄像头：单目、双目、多目。
- 惯性传感器（英文叫IMU，包括陀螺仪、加速度计）：智能手机标配。

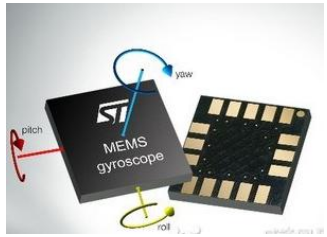
激光雷达

常见的单目摄像头

普通手机摄像头也可作为传感器

双目摄像头

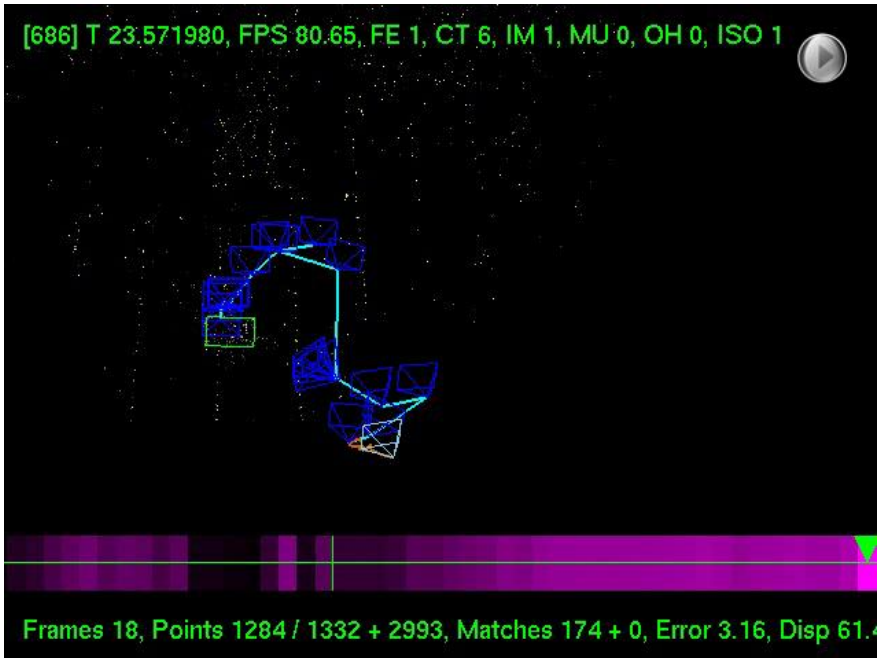
微软Kinect彩色-深度（RGBD）传感器



手机上的惯性传感器（IMU）

SLAM的运行结果

- 设备根据传感器的信息
  - 计算自身位置（在空间中的位置和朝向）
  - 构建环境地图（稀疏或者稠密的三维点云）



稀疏SLAM

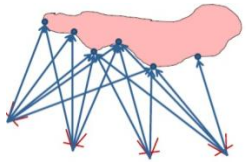
稠密SLAM

视觉SLAM系统常用的框架

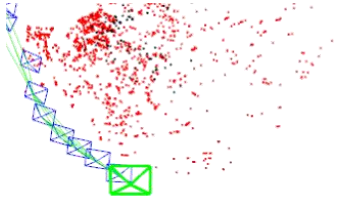
前台线程

视频流

输入传感器数据

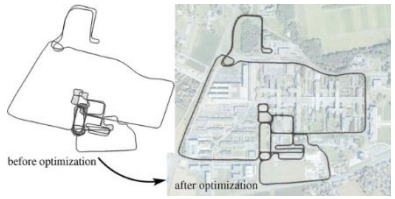


- SLAM初始化
- 特征跟踪与位姿实时求解



输出设备实时位姿和三维点云

后台线程



进行局部或全局优化，减少误差累积

场景回路检测

场景重定位

初始化

- SfM
  - 有时需要处理相机内参未知的情况，采用自定标等技术来估计出相机内参
- SLAM
  - 相机内参通常会预先标定好，固定不变
  - 单目SLAM初始化与SfM在相机内参已知情况下较类似（如Nistér的五点法[Nistér，2004]来做初始化）
  - 双目或者多目SLAM的初始化较为简单

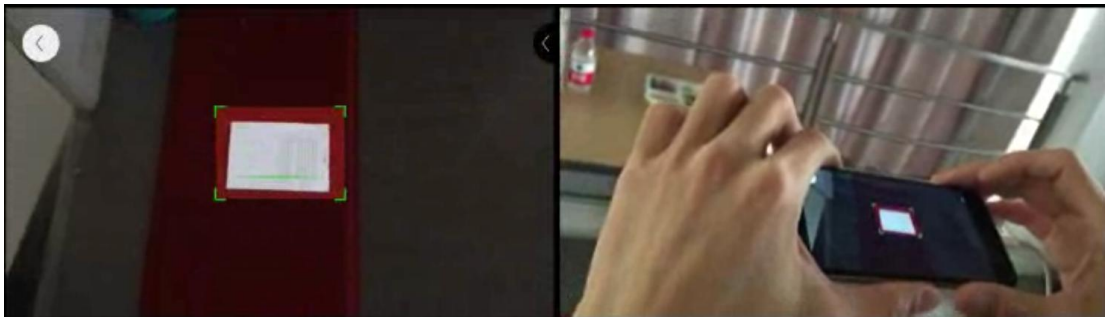
单目SLAM初始化

- 需求解的三维点和相机位姿相互依赖
  - 三角化三维点：需要相对位姿
  - 相对位姿：需要场景三维点
- 流程
  - 根据两帧间2D-2D的特征匹配估计相机初始位姿
  - 三角化出匹配的特征点的三维位置
  - 使用3D-2D的方法对这些特征点进行跟踪匹配，并求解相机运动

单目SLAM初始化的常见策略

- PTAM：需要用户指定两个关键帧
- ORB-SLAM：自动选取两帧，同时估计单应性矩阵（平面场景）和基础矩阵，选其中一个用于初始化
- RKSLAM：
  - 单帧初始化，假设相机正对着一个距离固定的平面，并使用了常数的深度来初始化三维坐标
  - 根据已知尺寸标志物来估计初始平面深度信息

PTAM初始化



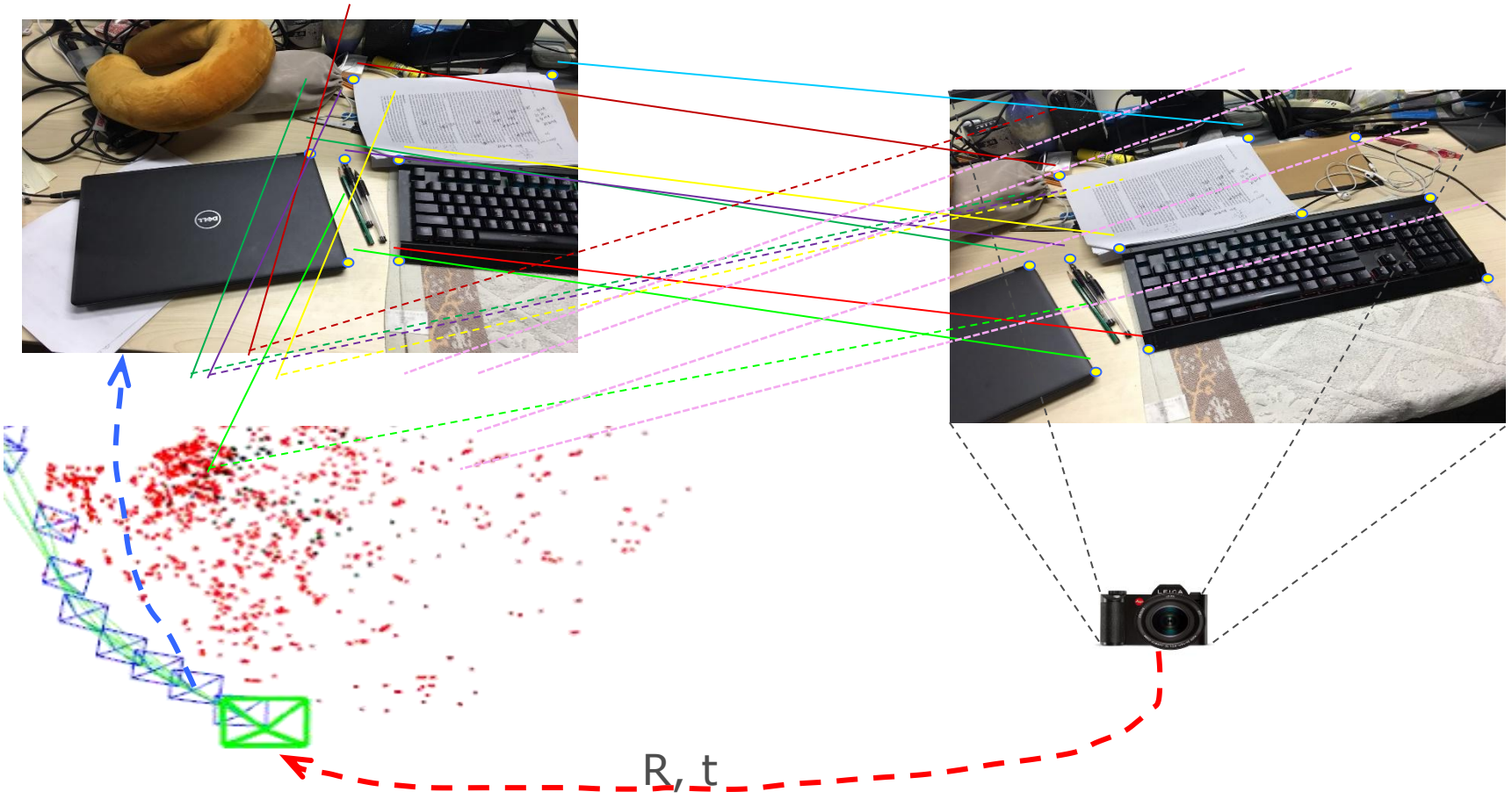
RKSLAM基于Marker初始化

多目SLAM初始化

- 优势
  - 预先标定多个同步相机之间的相对位姿变换
  - 三维信息可以通过多相机视图进行特征匹配并三角化来获得
  - 相机运动直接通过3D-2D的方法求解
- 实例
  - 双目SLAM：通过双目立体匹配（Stereo Matching）获得左右视图上的匹配点的视差，利用相似三角形来计算得到三维点坐标
  - 多目SLAM：初始化过程跟双目SLAM类似，无非是在更多的相机视图之间进行特征匹配并三角化出特征点的三维坐标

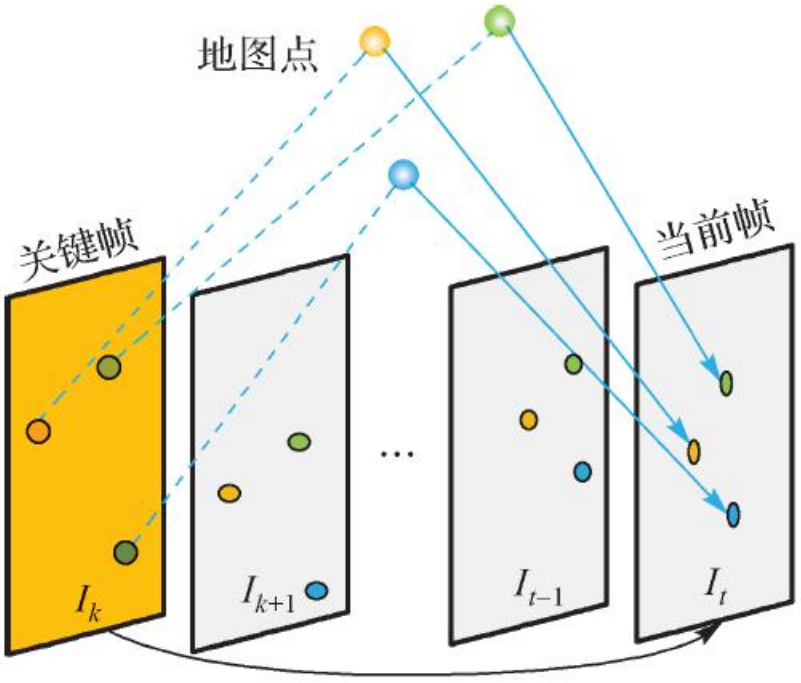
前台跟踪

特征检测 → 特征匹配 → 3D-2D运动估计 → 地图扩展



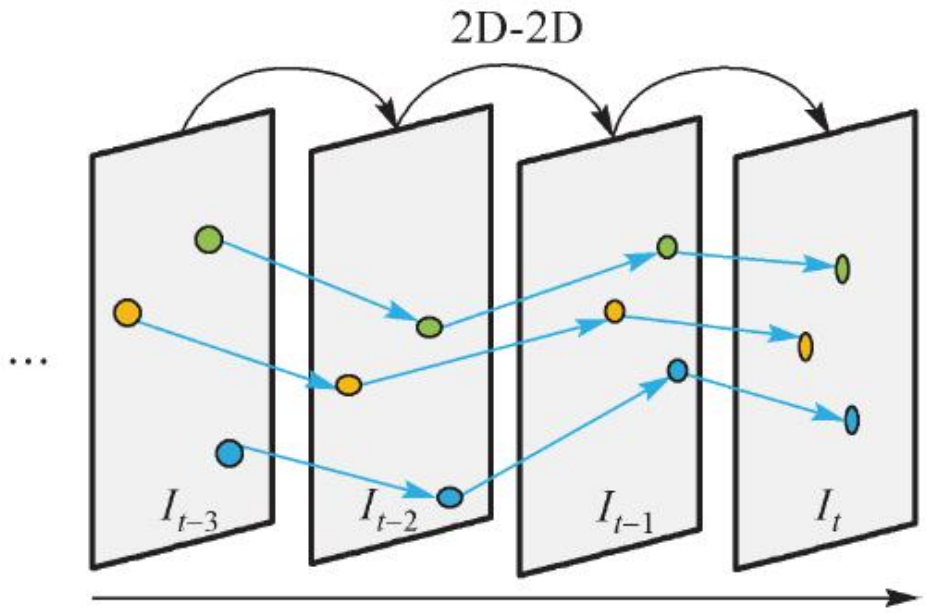
前台跟踪

- 关键帧匹配
  - 当前帧的特征点的最终匹配结果是同关键帧的点建立关联
  - 特征匹配过程一般会利用运动先验得到初始位置，减小搜索窗口大小，加快匹配速度
  - 特征提取及匹配过程需要考虑空间分布均匀

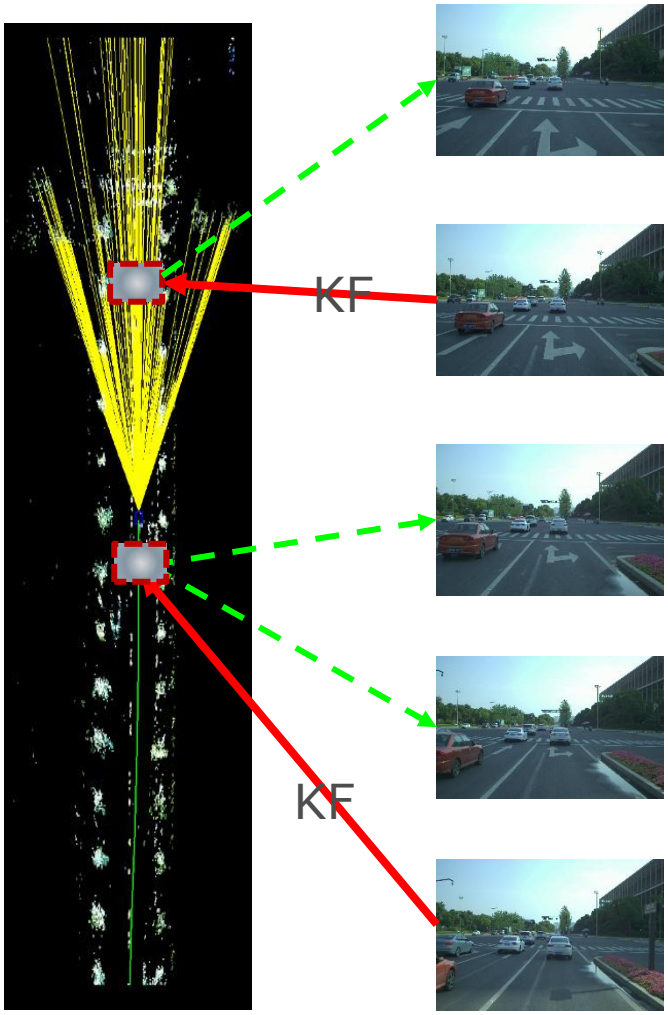


前台跟踪

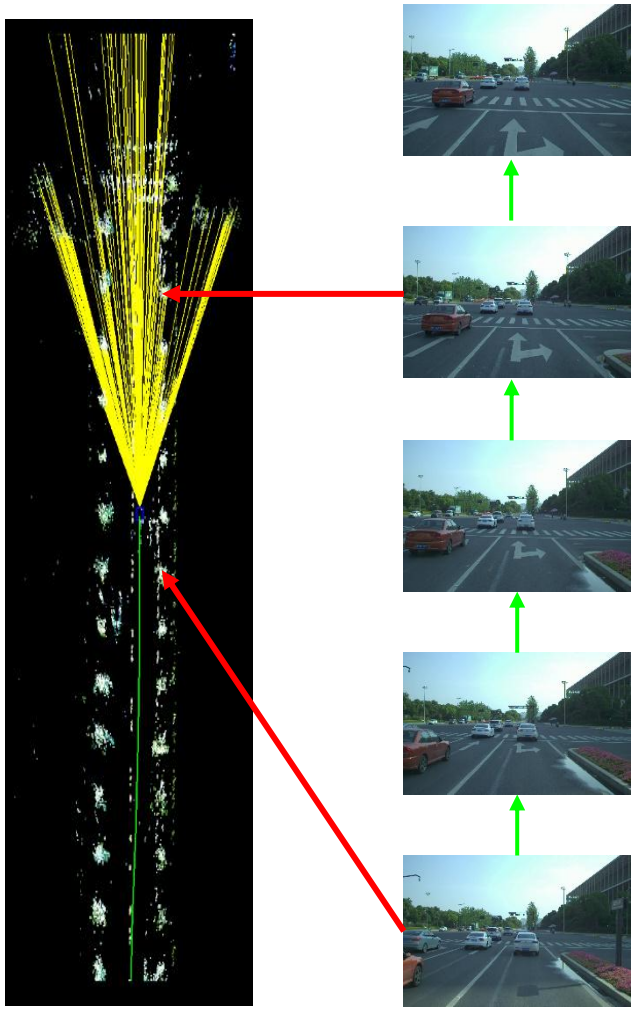
- 连续帧跟踪
  - 特征匹配基于光流法（光度误差），而非描述符匹配
  - 特征在时空上需要保持连续性，如果发生中断则认为是新的特征
  - 三维点的更新频率更高，需要能够快速完成三角化
  - 运动较快时容易LOST，容易产生累积误差



前台跟踪



关键帧匹配



连续跟踪

前台跟踪

- 运动估计 $\arg\min_{R,t} \sum_i \sum_j \left\| p_i^j - \hat{p}_i^j \right\|$

其中，p_i^j 表示在第i帧图像上与三维点*j*对应的检测到的特征点

$\hat{p}_i^j = \Pi(K_i(R_i X_j + t_i))$ 表示三维点*j*在第i帧图像上的投影点

  - DLT：构造一个包含12个未知数的增广矩阵进行线性求解，近似解！
  - P3P: 需要3对3D-2D对应关系，以及一组额外的点对进行验证
  - EPnP: 4对不共面的3D-2D点对
  - …

后台跟踪

- 前台跟踪获得相机状态或地图状态的初值等作为后端优化的输入
  - 实时性要求高的模块放在前端
  - 运算任务重的模块放在后端
    - Bundle Adjustment优化
    - 回路闭合
    - 重定位
    - 稠密三维恢复

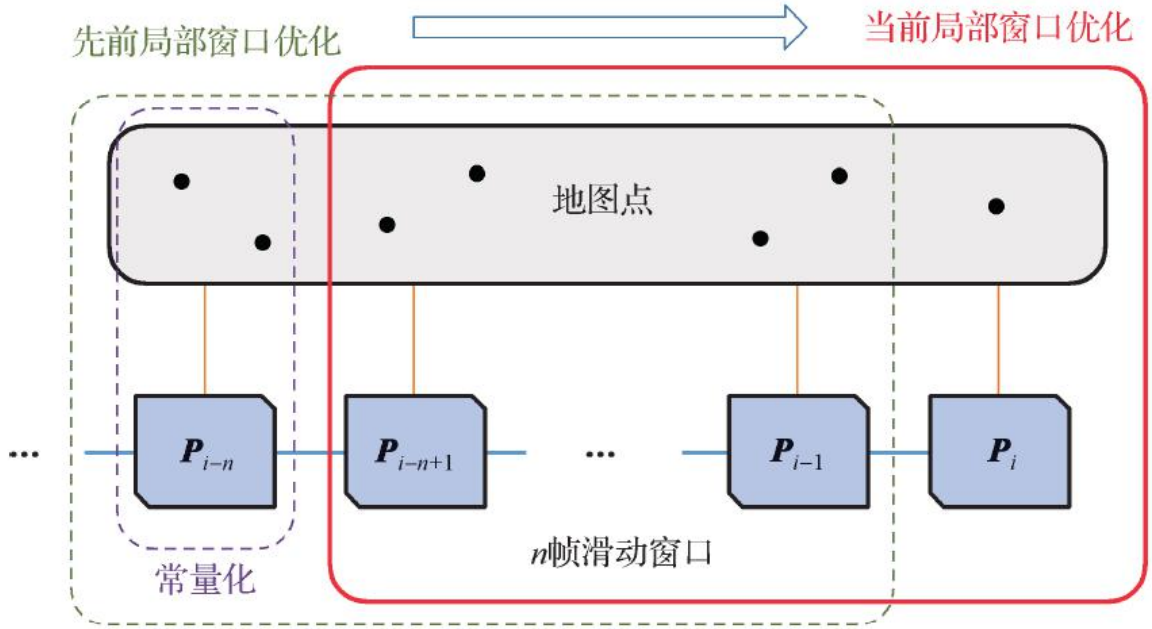
后台跟踪

- 后端优化方法
  - 全局优化：对所有历史相机状态和地图状态进行批量式优化；使用了所有信息，精度最高，速度慢。
  - 局部窗口优化：采用一个滑动窗口方式进行，最新状态被加入滑动窗口进行优化，最旧状态被移除，窗口内始终保持一定数量的状态；只考虑历史信息，速度快，精度相对较低。
  - 带状态先验的局部窗口优化：状态滑出时，对留在窗口中的状态做边缘化，其结果作为状态先验加入到下一次窗口优化；保留了历史信息，速度较快，精度次于全局优化。

后台跟踪

- 局部窗口优化
  - 滑动窗口：最新的一帧图像加入滑动窗口时，选择一帧移出滑动窗口，保持窗口大小不变
  - 窗口内的状态 $\mathcal{J} \triangleq I_t, I_{t+1}, I_{t+2}, \cdots, I_{t+k}$ 最小化重投影误差

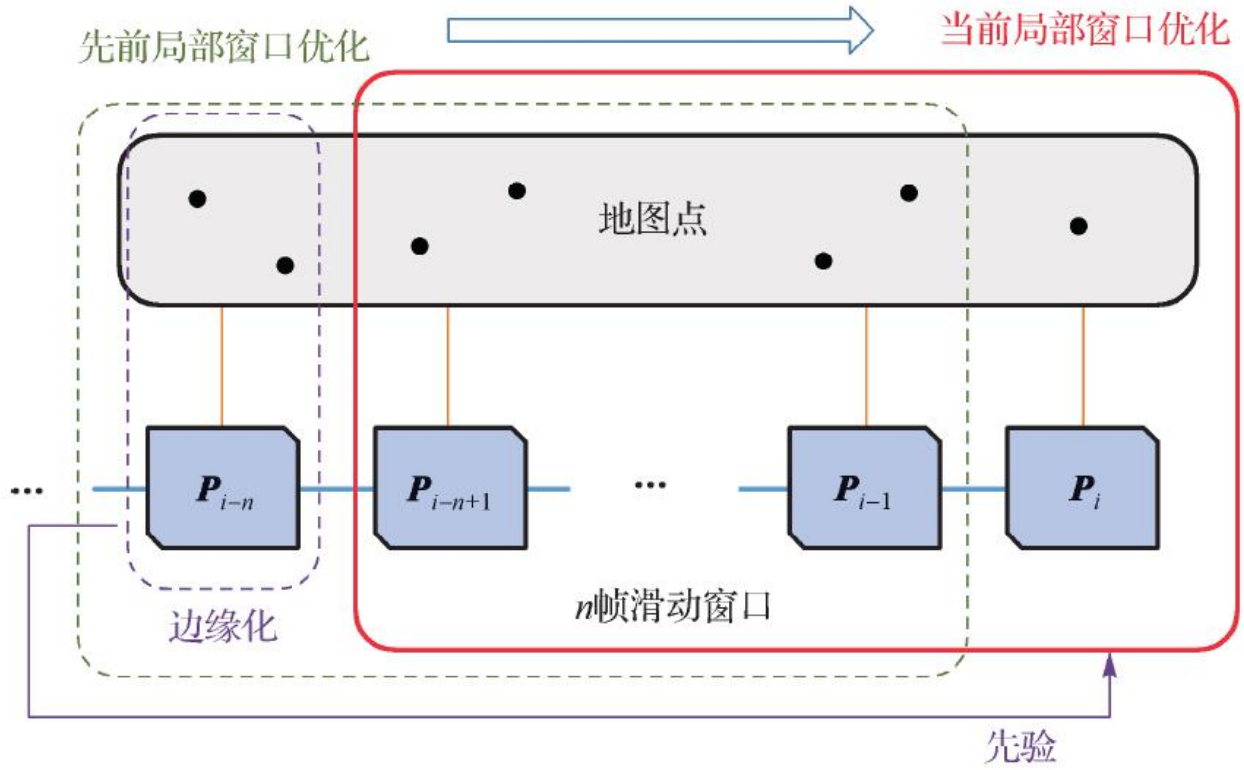
$$\arg\min_{\mathcal{J}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathcal{J}_i} \left\| \Pi\left(\mathrm{K}_i\left(\mathrm{R}_i \mathbf{p}_j + \mathbf{t}_i\right)\right) - \mathbf{x}_{ij} \right\|_{\Sigma_{ij}}^2$$



后台跟踪

- 带状态先验的局部窗口优化
  - 每次将状态和相应的三维点滑出窗口时，对其做边缘化

$$\arg\min_{\boldsymbol{C},\boldsymbol{M}}\left(\| \boldsymbol{r}_{\mathcal{I}} \|^2_{\Sigma_{\mathcal{I}}} + \sum_{k=i-n+1}^{i}\sum_{\boldsymbol{x}_{kj}\in\boldsymbol{I}_k} \| \pi(\boldsymbol{K}_k(\boldsymbol{R}_k\boldsymbol{X}_j + \boldsymbol{t}_k)) - \boldsymbol{x}_{kj} \|^2_{\Sigma_{kj}}\right)$$

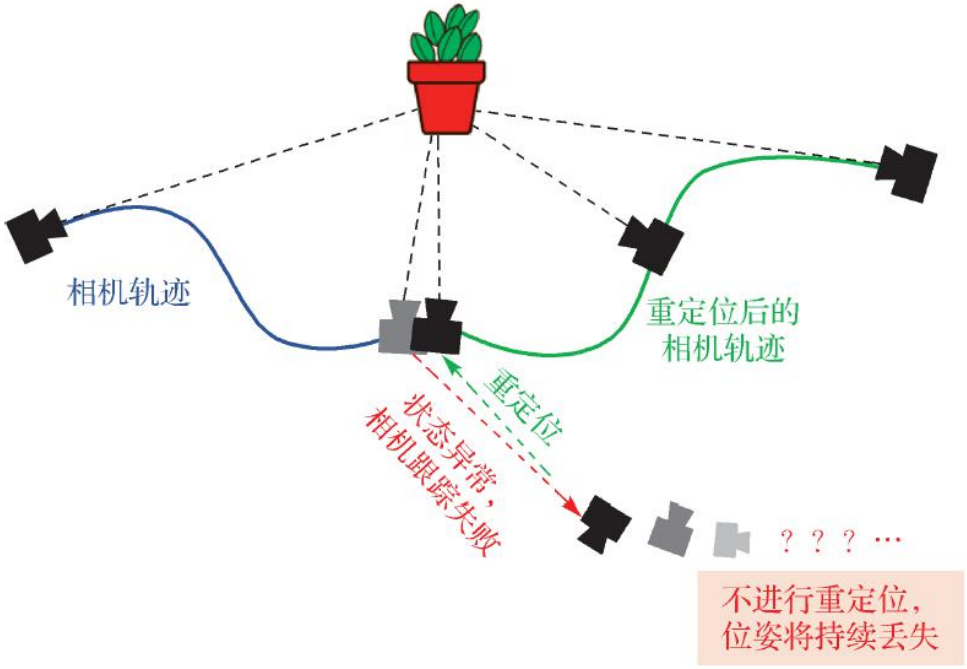


后台跟踪

- 全局优化
  - 对所有历史相机状态和地图状态进行批量式优化，消除误差累积
  - 优化时机
    - 检测到回路闭合
    - 定时触发
  - 保证后台地图能够在可接受的时间内完成
    - 无结构的建模，只求解位姿图优化问题
    - 增量式的集束调整方法（iSAM/iSAM2，EIBA/ICE-BA, SLAM++）
  - 状态删除的策略（确保地图不会无限制的增长）
    - 直接删除：保证运行时间上限，但丢失了信息，精度较低
    - 边缘化后再删除：保留了部分信息，稀疏性变差，精度较高

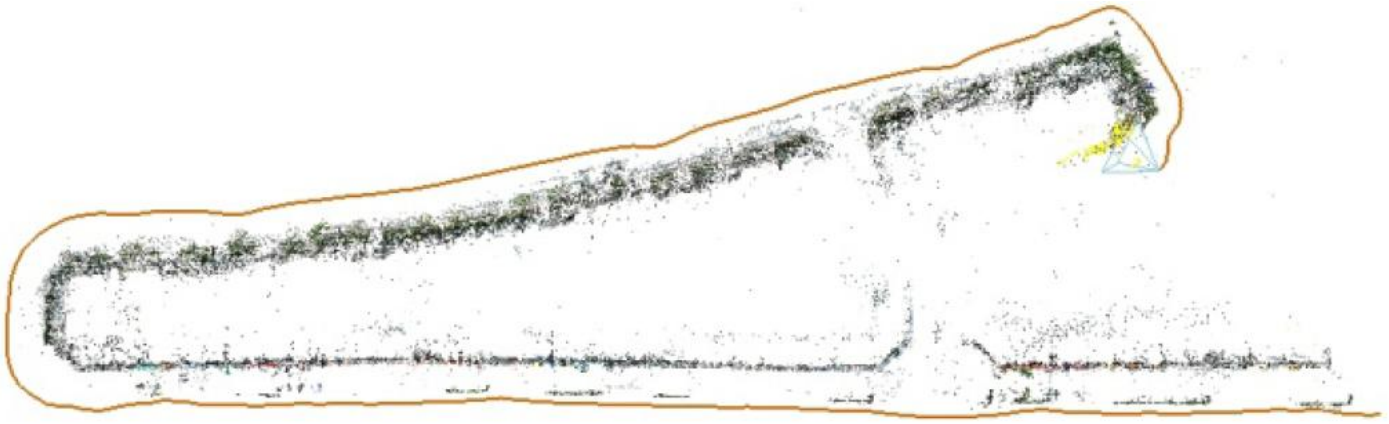
重定位

- 必要性
  - 视觉 SLAM有时会出现跟踪失败的情况，图像质量过差或者图像内容缺少特征都可能导致一段时间内跟踪失败。
  - 需要从相机跟踪失败的状态恢复到正常跟踪状态。
- 目标
  - 在连续跟踪失败的情况下快速重新得到当前的精确位姿



回路闭合

- 必要性
  - 随着时间和运动距离的加长，相机跟踪误差会累积到一个较大的量级。
  - 通过闭合回环约束相机轨迹，有效消除累积误差



(a) 回路闭合之前的结果

(b) 回路闭合之后的结果

重定位与回路闭合

- 相似性
  - 初始操作类似：寻找当前场景与已经生成的地图的联系，找到曾经访问过的场景。
  - 本质上都是一个图像检索的过程。
- 优化目标不同
  - 重定位只需得到相机的当前位姿
  - 回路闭合需要修正整个回路的轨迹以及相关的三维点坐标

重定位与回路闭合

- 图像检索难点
  - 随着场景的拓展，SLAM 系统中的关键帧数目会持续增长；
  - 如何快速、精准地从大量图像中找到与当前帧最相似的帧是图像检索模块的关键。
- 图像检索
  - 局部特征检索方法
  - 全局图像检索方法

局部特征检索方法

- 基于各种局部特征点的检索方法
  - 可以在一定程度上容忍视角的变化
  - 对重复纹理和图像模糊的容忍度较差

www.cs.washington.edu

全局图像检索方法

- 使用整张图像信息进行检索。
  - 以Gist为代表的传统方法的缺点是速度慢，对于视角变化的容忍度差。
  - 基于深度学习方法需要大量的数据进行预训练，这种方法能直接从图像中得到相机姿态。

PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization

原图（上）点云图（中）使用神经网络得到姿态将点云图映射到原图（下）

SLAM方法分类

- Filter-based SLAM
  - Davison et al.2007 (MonoSLAM), Eade and Drummond 2006, Mourikis et al. 2007 (MSCKF), …
- Keyframe-based SLAM
  - Klein and Murray 2007,2008 (PTAM), Castle et al.2008, Tan et al. 2013 (RDSLAM), Mur-Artal et al. 2015 (ORB-SLAM), Liu et al. 2016 (RKSLAM), …
- Direct Tracking based SLAM
  - Engel et al. 2014 (LSD-SLAM), Forster et al. 2014 (SVO), Engel et al. 2018 (DSO)

Extended Kalman Filter

- State at time k, model as multivariate Gaussian

$$x_k \sim N(\hat{x}_k, P_k)$$

mean covariance

- State transition model

$$x_k = f(x_{k-1}) + w_k$$

$$w_k \sim N(0, Q_k)$$ Process noise

- State observation model

$$z_k = h(x_k) + v_k$$

$$v_k \sim N(0, R_k)$$ Observation noise

Extended Kalman Filter

- Predict

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1})$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

$$F_k = \partial f / \partial x \big|_{\hat{x}_{k-1|k-1}}$$

- Update

$$S_k = H_k P_{k|k-1} H_k^T + R_k$$ Innovation covariance

$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$

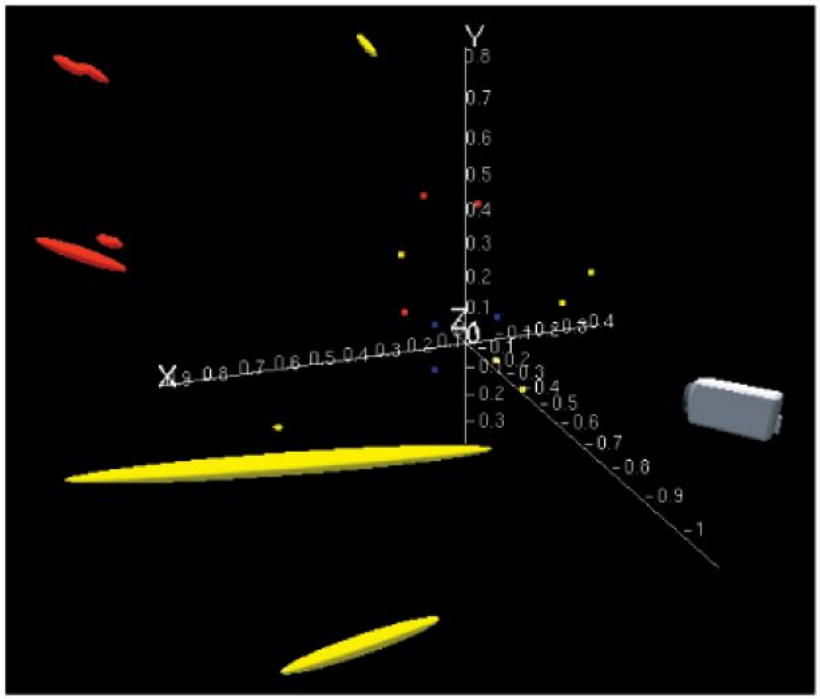
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - h(\hat{x}_{k|k-1}))$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

$$H_k = \partial h / \partial x \big|_{\hat{x}_{k|k-1}}$$

MonoSLAM

- Map representation



$$x = \begin{pmatrix} C \\ X \end{pmatrix} = \begin{pmatrix} C \\ X_1 \\ X_2 \\ \vdots \end{pmatrix}$$

camera state

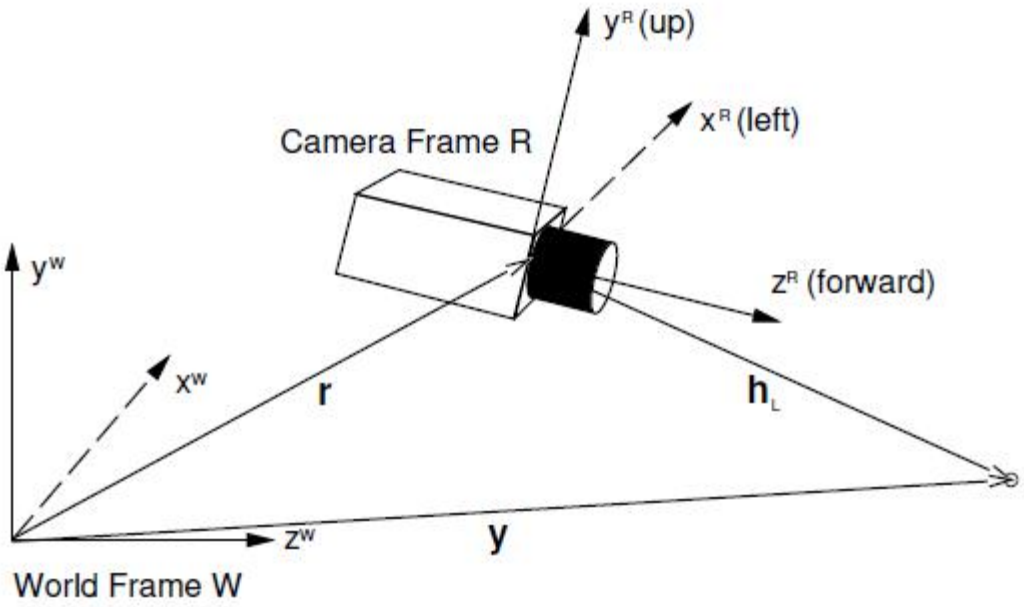
point state

$$P = \begin{pmatrix} P_{CC} & P_{CX_1} & P_{CX_2} & \cdots \\ P_{X_1C} & P_{X_1X_1} & P_{X_1X_2} & \cdots \\ P_{X_2C} & P_{X_2X_1} & P_{X_2X_2} & \cdots \\ \vdots & \vdots & \vdots & \end{pmatrix}$$

A. J. Davison, N. D. Molton, I. Reid, and O. Stasse. MonoSLAM: Real-time single camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 29(6):1052-1067, 2007.

MonoSLAM

- Camera state



$$C_k = \begin{pmatrix} p_k \\ q_k \\ v_k \\ \omega_k \end{pmatrix}$$

camera position

orientation quaternion

linear velocity

angular velocity

MonoSLAM

- Predict

$$w_k = \begin{pmatrix} a_k \\ \alpha_k \end{pmatrix}$$

linear acceleration

angular acceleration

$$w_k \sim N\left(0, \mathrm{diag}(Q_a, Q_\alpha)\right)$$

$$C_k = \begin{pmatrix} p_k \\ q_k \\ v_k \\ \omega_k \end{pmatrix} = \begin{pmatrix} p_{k-1} + (v_{k-1|} + a_k)\Delta t \\ q((\omega_{k-1} + \alpha_k)\Delta t) \otimes q_{k-1} \\ v_{k-1|} + a_k \\ \omega_{k-1} + \alpha_k \end{pmatrix}$$

$$X_k = X_{k-1}$$

MonoSLAM

- Predicted features position

$$z_i = \pi(X_i, C) + v_i$$

$$v_i \sim N(0, R)$$

- Innovation covariance
  - Elliptical feature search region

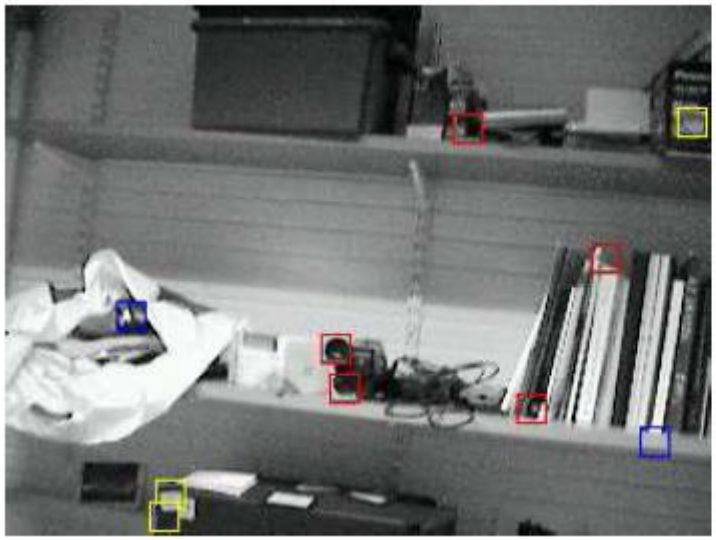
$$S_i = J_C P_{CC} J_C^T + J_C P_{CX_i} J_{X_i}^T + J_{X_i} P_{X_i C} J_C^T + J_{X_i} P_{X_i X_i} J_{X_i}^T + R$$

$$J_C = \frac{\partial z_i}{\partial C}$$

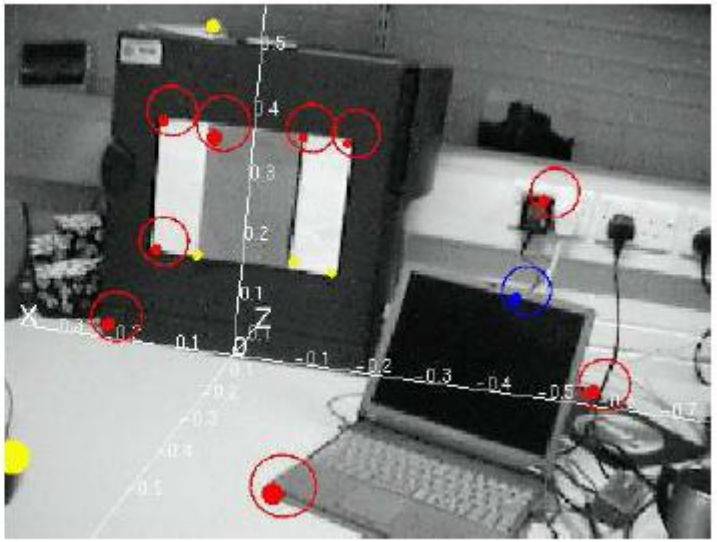
$$J_{X_i} = \frac{\partial z_i}{\partial X_i}$$

MonoSLAM

- Active search



Shi and Tomasi Feature



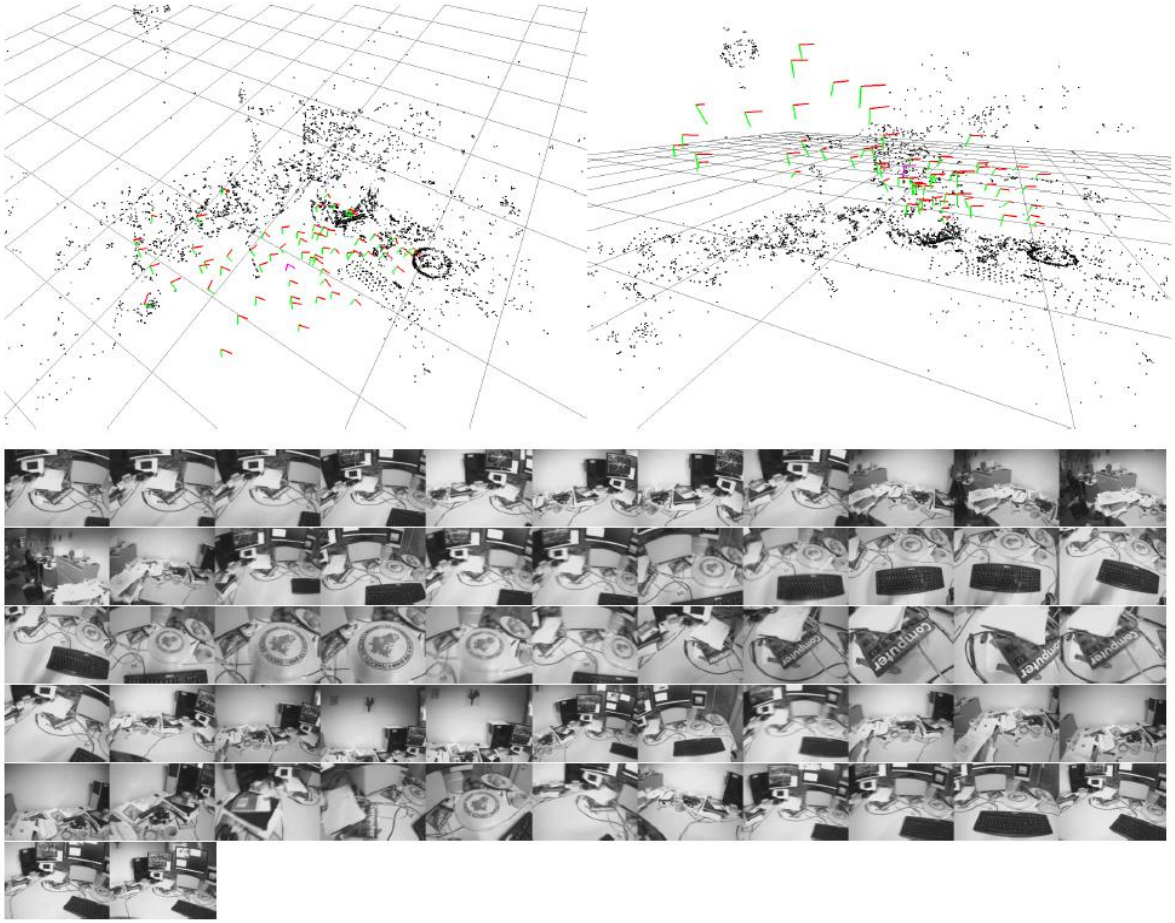
Elliptical search region

MonoSLAM

- Complexity
  - $O(N^3)$ per frame
- Scalability
  - Hundreds of points

PTAM: Parallel Tracking and Mapping

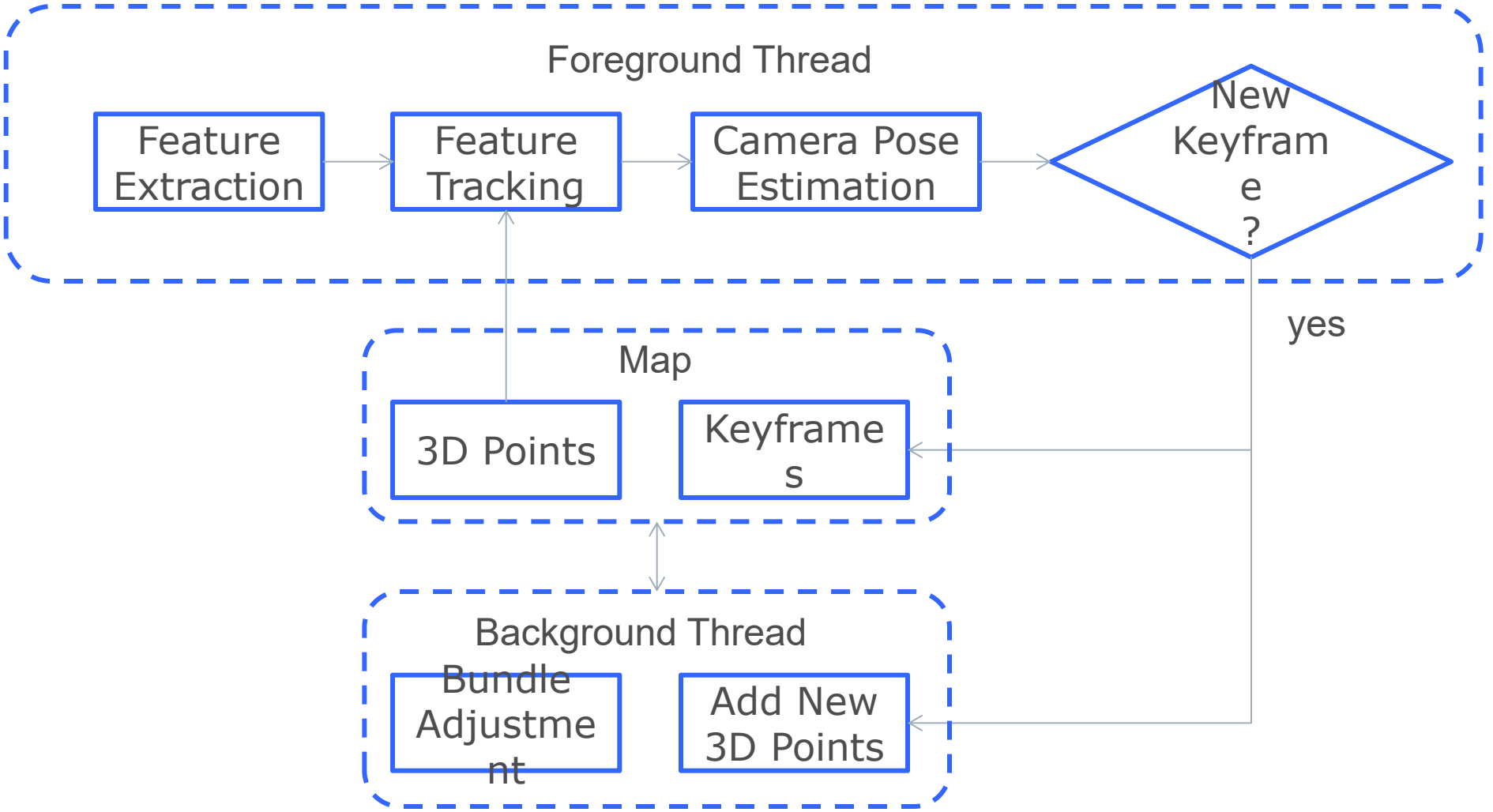
- Map representation



G. Klein and D. W. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR), 2007.

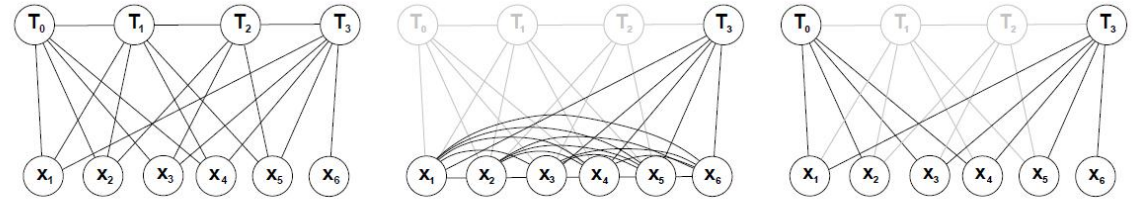
PTAM: Parallel Tracking and Mapping

- Overview

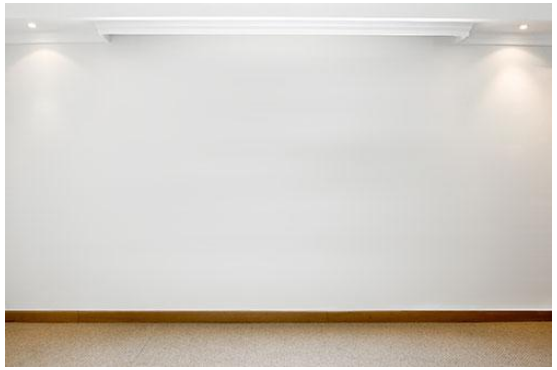


Keyframe-based SLAM vs Filtering-based SLAM

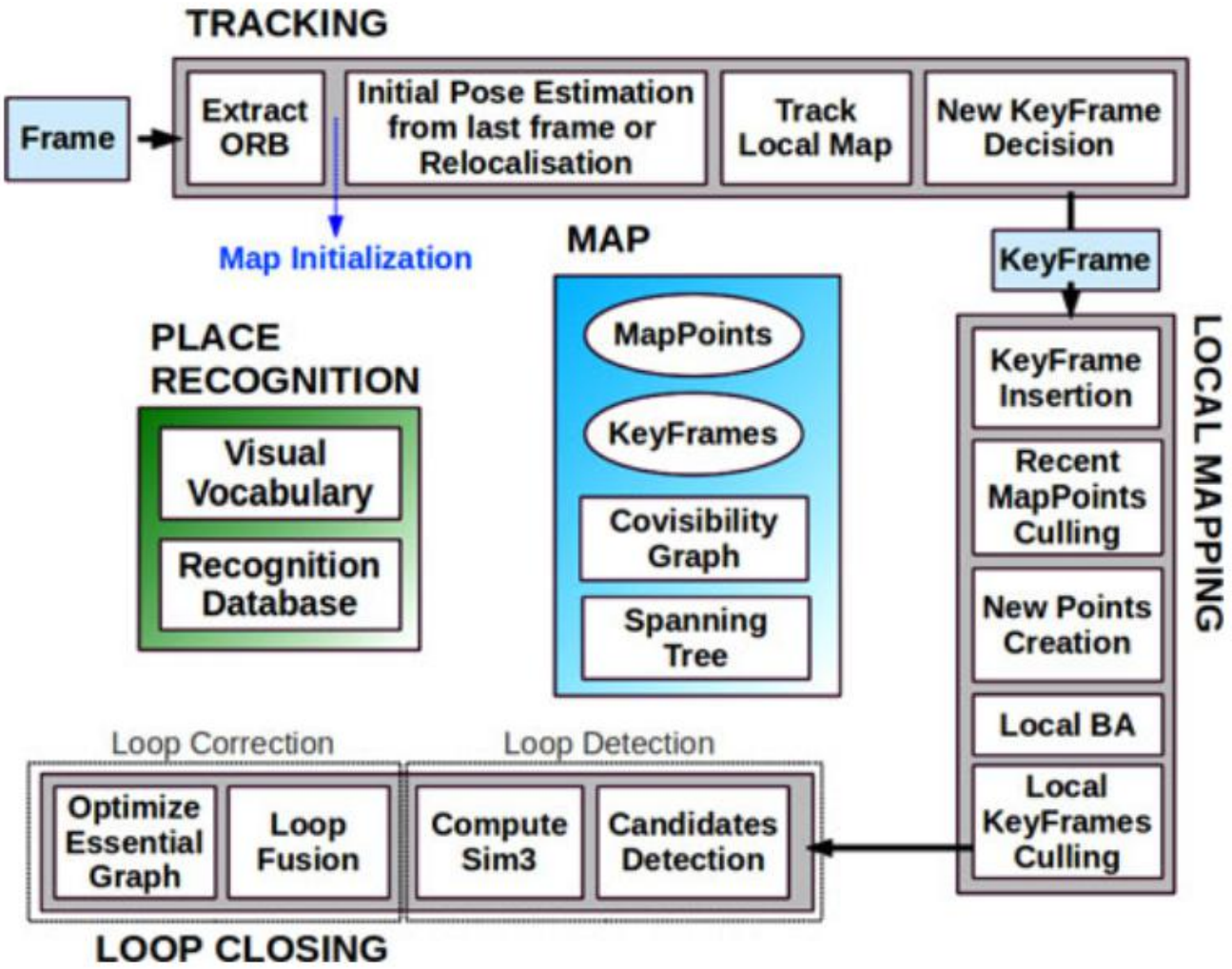
- Advantages
  - Accuracy
  - Efficiency
  - Scalability
- Disadvantages
  - Sensitive to strong rotation
- Challenges for both
  - Fast motion
  - Motion blur
  - Insufficient texture



H. Strasdat, J. Montiel, and A. J. Davison. Visual SLAM: Why filter? Image and Vision Computing, 30:65-77, 2012.



ORB-SLAM

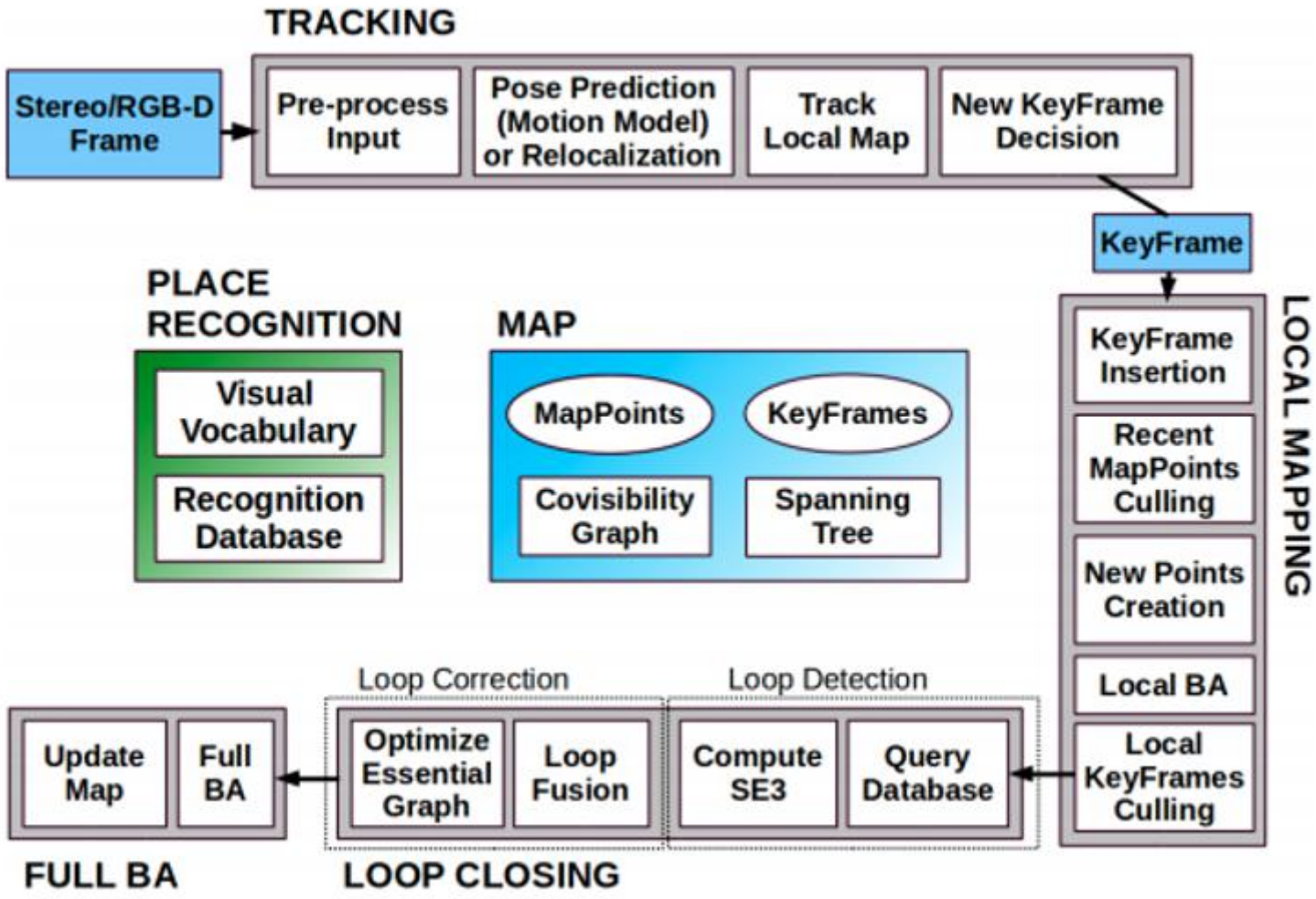


Raul Mur-Artal, J. M. M. Montiel, Juan D. Tardós: ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Trans. Robotics 31(5): 1147-1163 (2015).

ORB-SLAM

- 基本延续了 PTAM 的算法框架,但对框架中的大部分组件都做了改进
  - 选用ORB特征, 匹配和重定位性能更好.
  - 加入了循环回路的检测和闭合机制, 以消除误差累积.
  - 通过检测视差来自动选择初始化的两帧.
  - 采用一种更鲁棒的关键帧和三维点的选择机制.

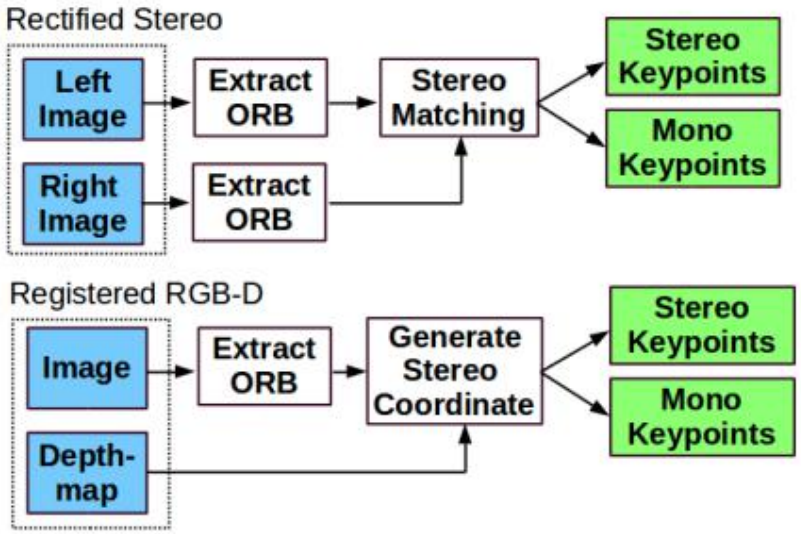
ORB-SLAM2

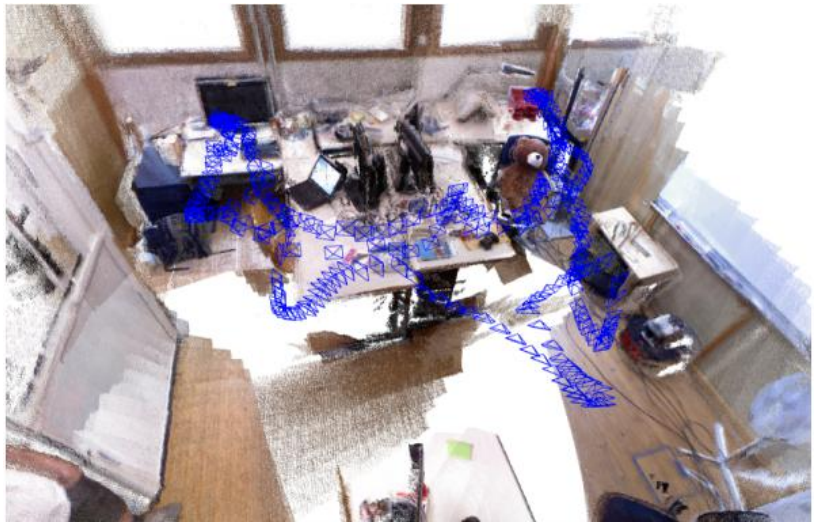


Mur-Artal R, Tardós J D. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras[J]. IEEE Transactions on Robotics, 2017, 33(5): 1255-1262.

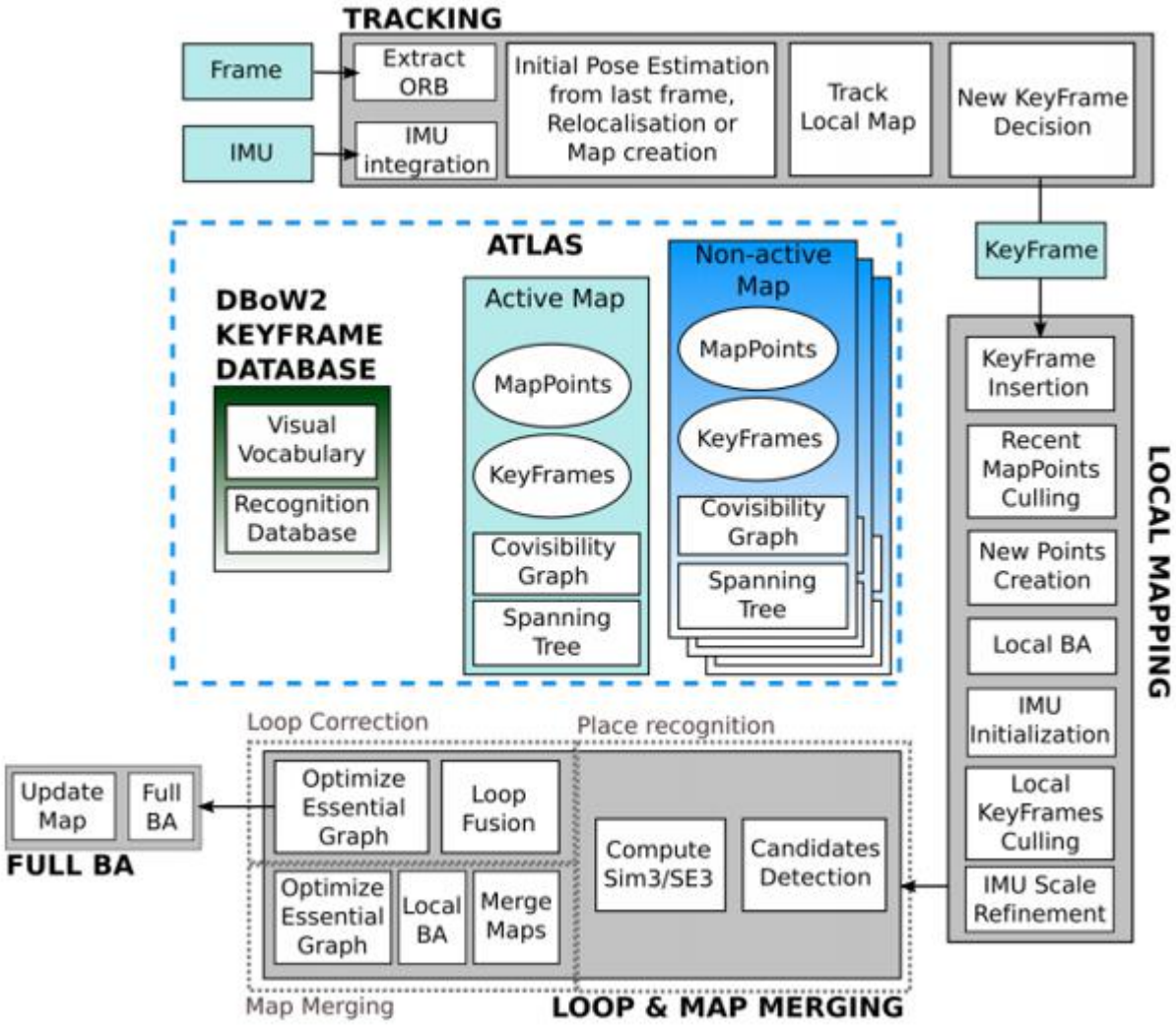
ORB-SLAM2

- 在ORB-SLAM的基础上做了改进
  - 同时支持单目、双目和RGB-D输入
  - 加入了全局BA线程，在检测到回路之后启用全局优化，得到全局最优解





ORB-SLAM3

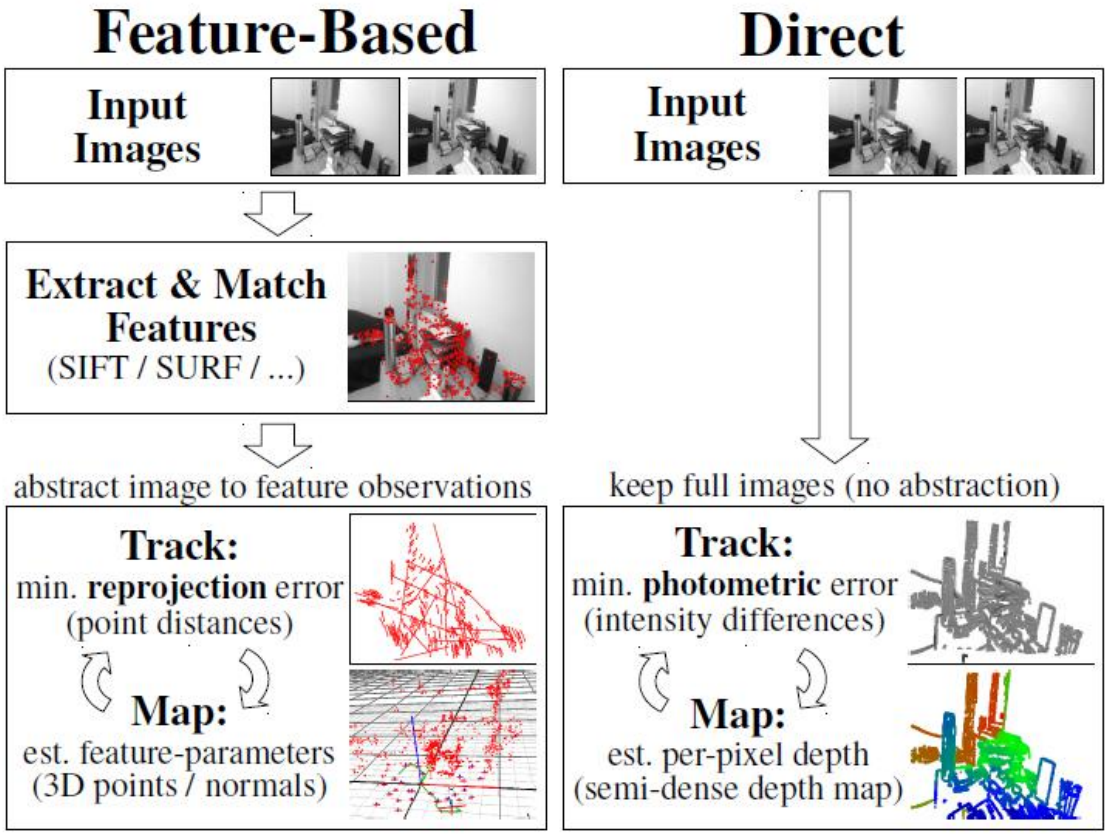


Campos C, Elvira R, Rodríguez J J G, et al. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. arXiv preprint arXiv:2007.11898, 2020.

ORB-SLAM3

- 在ORB-SLAM、ORB-SLAM2的基础上做了改进
  - 采用了抽象相机模型
  - 增加了IMU输入，采用最大后验估计的方式初始化和优化IMU
  - 采用了多地图机制，在跟踪丢失且无法重定位时创建新的地图，并在闭环检测线程中加入了地图融合机制

Direct Tracking



Thomas Schops, Jakob Engel, Daniel Cremers: Semi-dense visual odometry for AR on a smartphone. ISMAR 2014: 145-150.

Direct Tracking

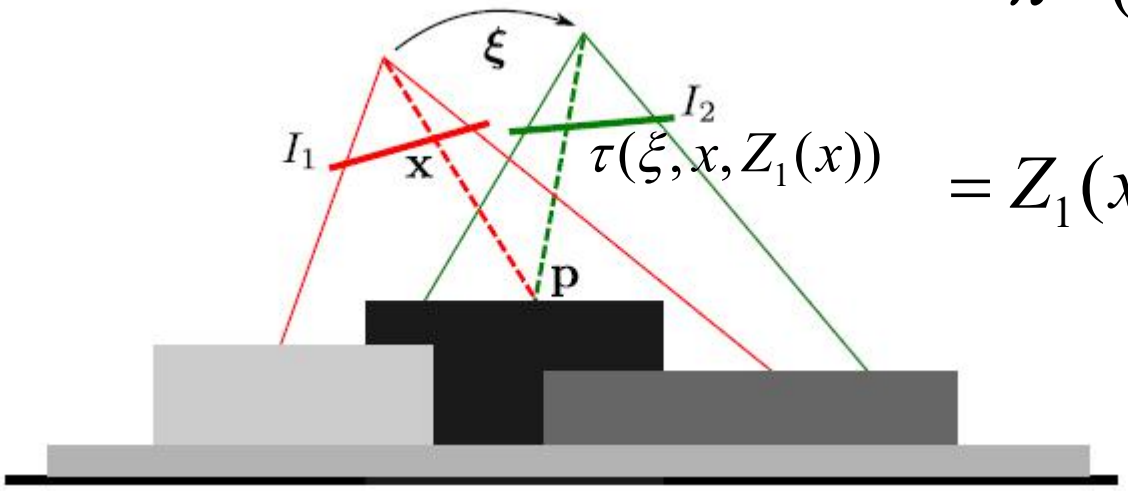
- Goal
  - Estimate the camera motion ξ by aligning intensity images I_1 and I_2 with depth map Z_1 of I_1
- Assumption

$$I_1(x) = I_2(\tau(\xi, x, Z_1(x)))$$

warping function: maps a pixel from I_1 to I_2

Direct Tracking

- Warping function



$$p = \pi^{-1}(x, Z_1(x))$$
$$= \pi^{-1}((u,v)^T, Z_1(x))$$
$$= Z_1(x)\left(\frac{u - c_x}{f_x}, \frac{v - c_y}{f_y}\right)^T$$

Christian Kerl, Jürgen Sturm, Daniel Cremers: Robust odometry estimation for RGB-D cameras. ICRA 2013: 3748-3754

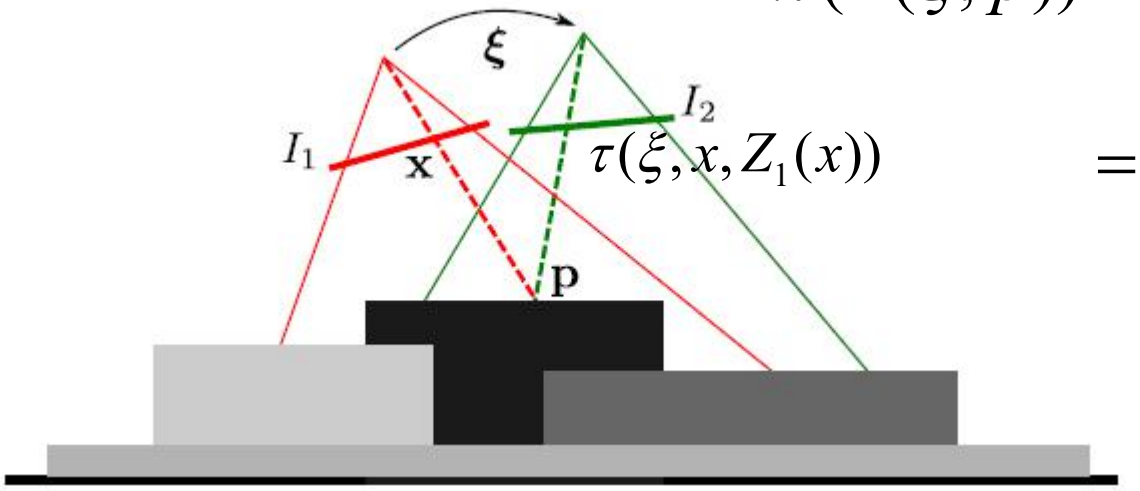
Direct Tracking

- Warping function

$$T(\xi, p) = Rp + t$$

$$\pi(T(\xi, p)) = \pi((X, Y, Z)^T)$$

$$= \left(\frac{f_x X}{Z} + c_x, \frac{f_y Y}{Z} + c_y \right)^T$$

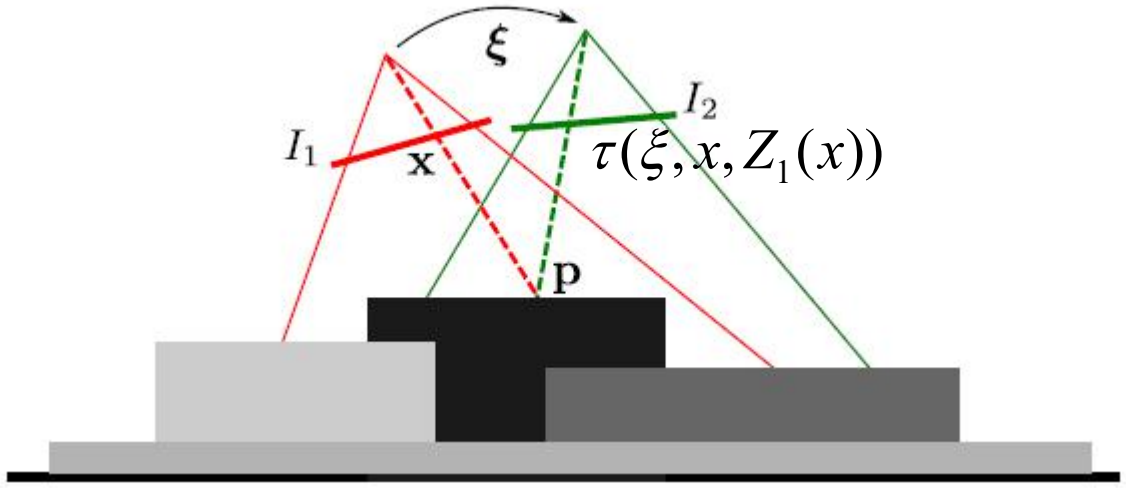


Christian Kerl, Jürgen Sturm, Daniel Cremers: Robust odometry estimation for RGB-D cameras. ICRA 2013: 3748-3754

Direct Tracking

- Warping function

$$\tau(\xi, x, Z_1(x)) = \pi(T(\xi, p))$$
$$= \pi(T(\xi, \pi^{-1}(x, Z_1(x))))$$



Christian Kerl, Jürgen Sturm, Daniel Cremers: Robust odometry estimation for RGB-D cameras. ICRA 2013: 3748-3754

Direct Tracking

- Residual of the *k*-th pixel

$$r_k(\xi) = I_2(w(\xi, x_k, Z_1(x_k))) - I_1(x_k)$$

- Posteriori likelihood

$$p(\xi \mid r) = \frac{p(r \mid \xi)p(\xi)}{p(r)} = \frac{\left(\prod_k p(r_k \mid \xi)\right)p(\xi)}{p(r)}$$

Semi-Dense Visual Odometry

Jakob Engel, Jürgen Sturm, Daniel Cremers: Semi-dense Visual Odometry for a Monocular Camera. ICCV 2013: 1449-1456

Semi-Dense Visual Odometry

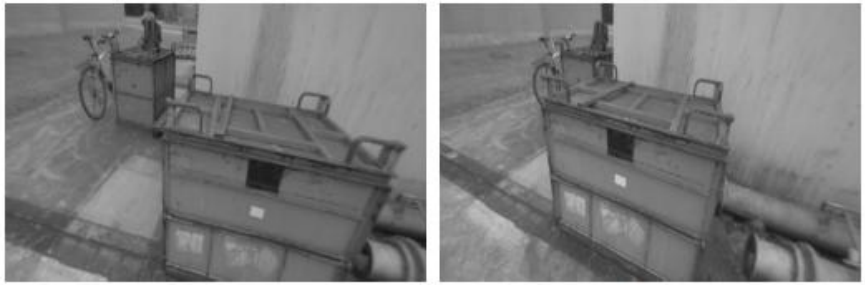
- Keyframe representation

$$K_i = (I_i, D_i, V_i)$$

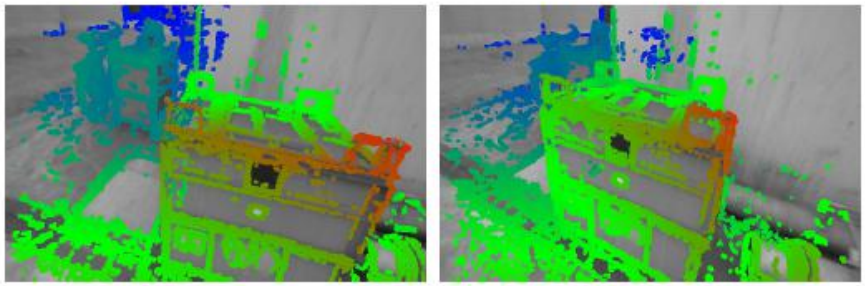
$i_i = I_i(x)$ image intensity

$d_i = D_i(x)$ inverse depth

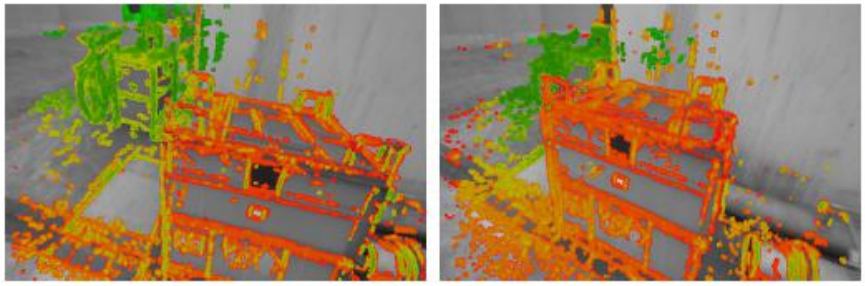
$\sigma_{d_i}^2 = V_i(x)$ inverse depth variance



(a) camera images I



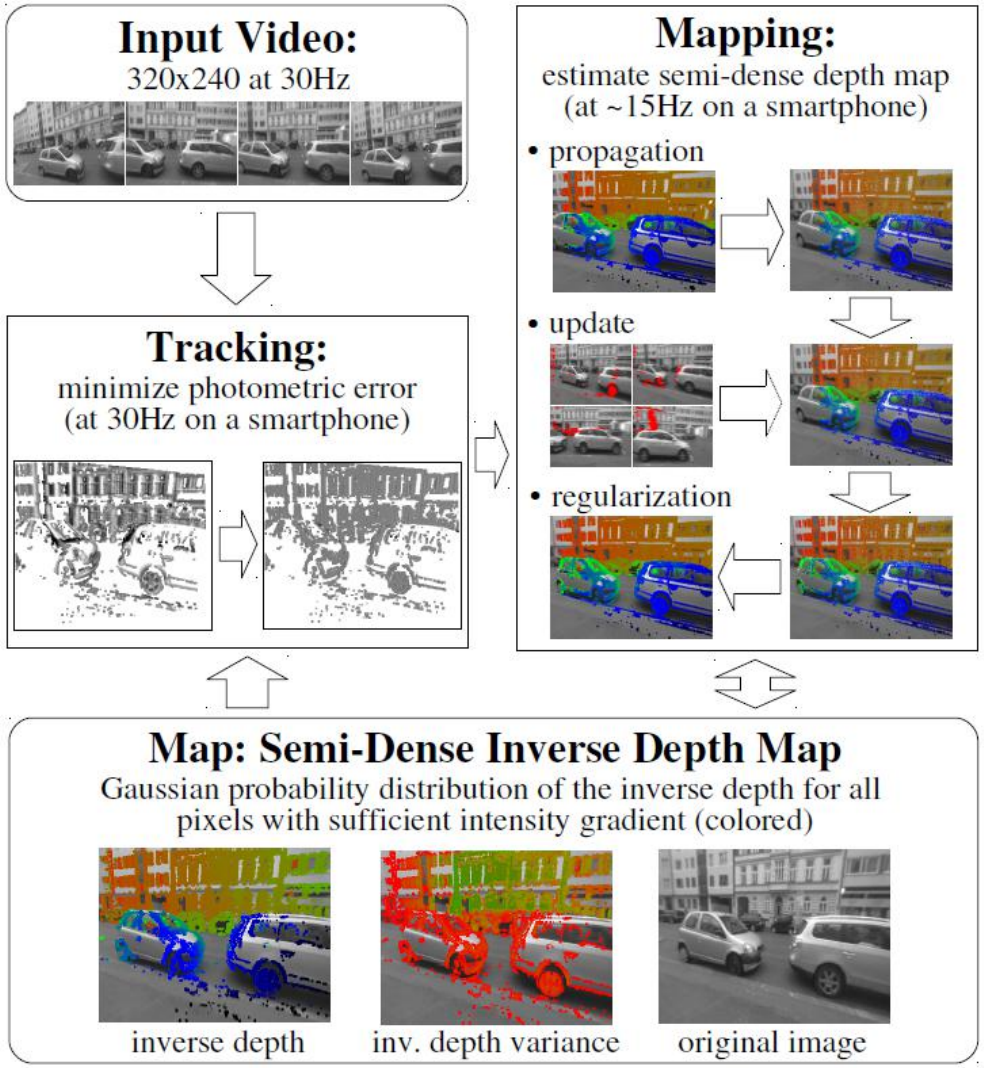
(b) estimated inverse depth maps D



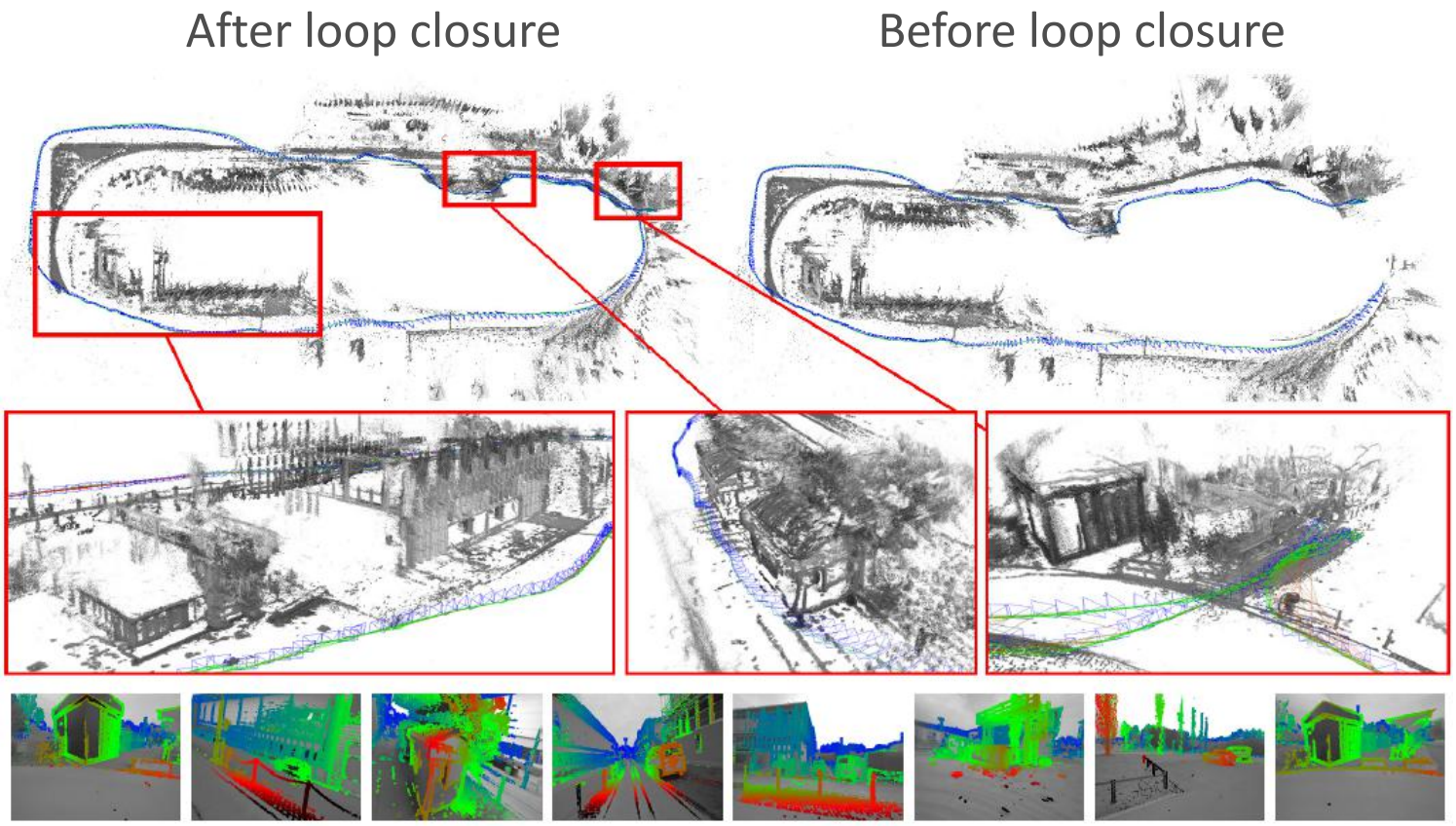
(c) inverse depth variance V

Semi-Dense Visual Odometry

- Overview



LSD-SLAM



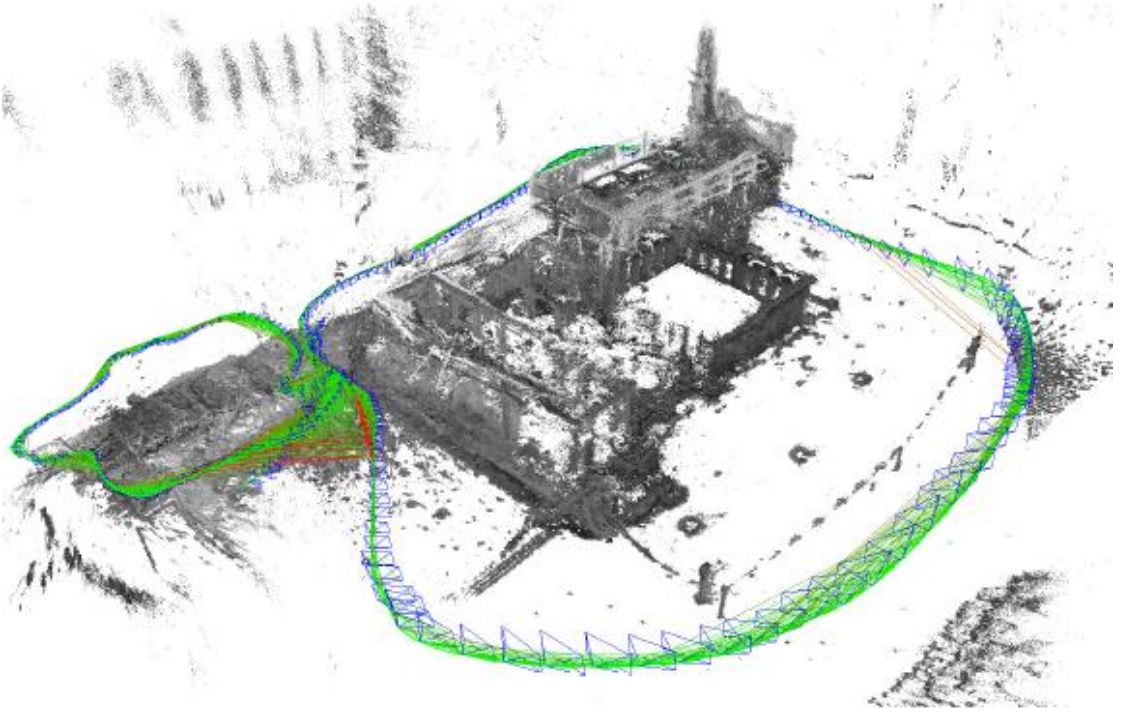
Jakob Engel, Thomas Schops, Daniel Cremers: LSD-SLAM: Large-Scale Direct Monocular SLAM. ECCV (2) 2014: 834-849.

LSD-SLAM

- Map representation
  - Pose graph of keyframes
  - Node: keyframe

    $K_i = (I_i, D_i, V_i)$
  - Edge: similarity transformation

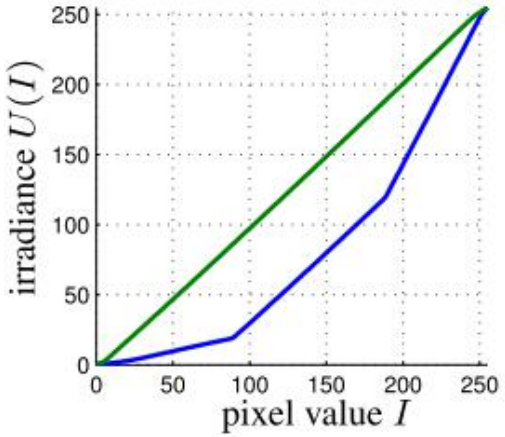
    $\xi_{ji} \in \mathrm{sim}(3)$



Direct Sparse Odometry

- Photometric calibration
  - Response calibration
  - Non-parametric Vignette Calibration
  - Exposure time

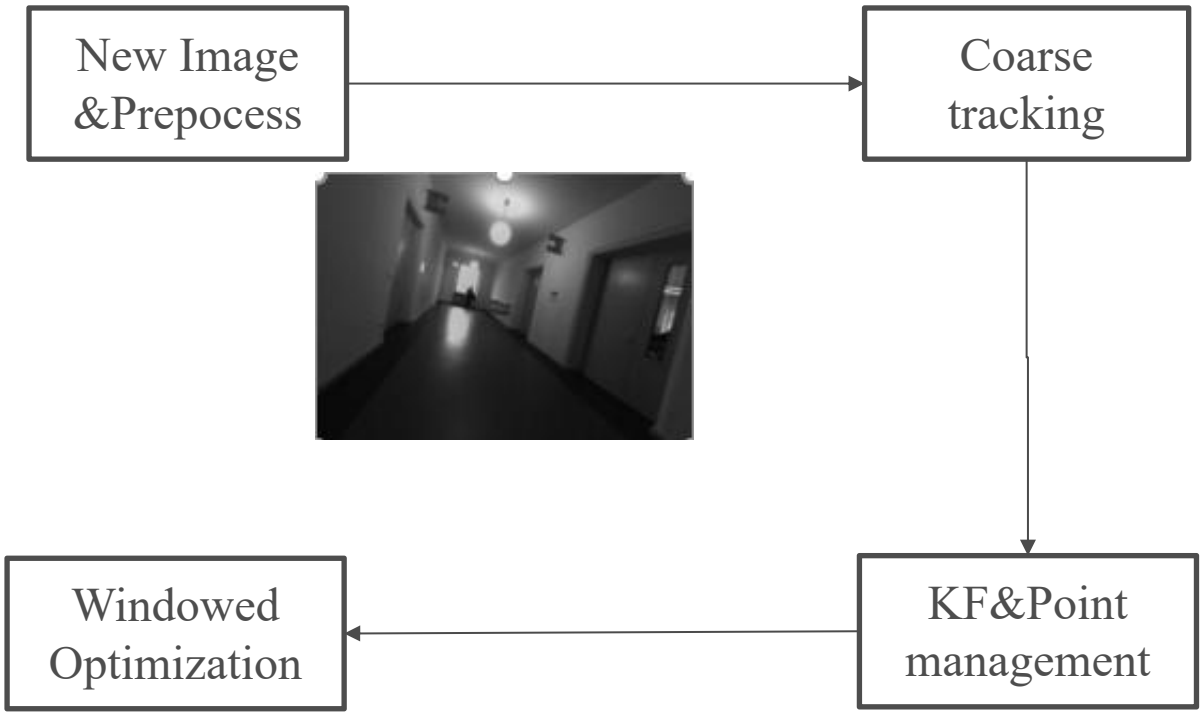
$$I_i(x) = G(t_i V(x) B_i(x))$$

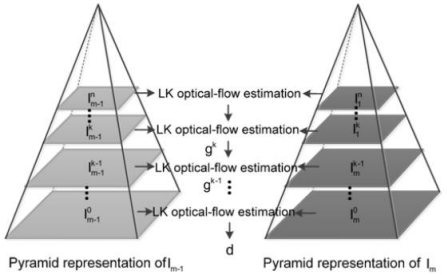


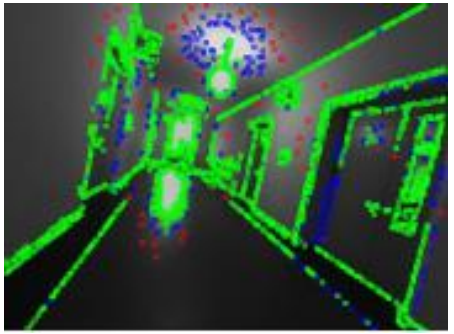
Jakob Engel, Vladlen Koltun, Daniel Cremers. Direct Sparse Odometry. IEEE Trans. Pattern Anal. Mach. Intell. 40(3): 611-625 (2018).

Direct Sparse Odometry

- Single thread pipeline







$$E_{pj} := \sum_{p \in N_p} w_p \left\| (I_j[p'] - b_j) - \frac{t_j e^{a_j}}{t_i e_{a_i}} (I_i[p] - b_i) \right\|_r$$

谢谢！