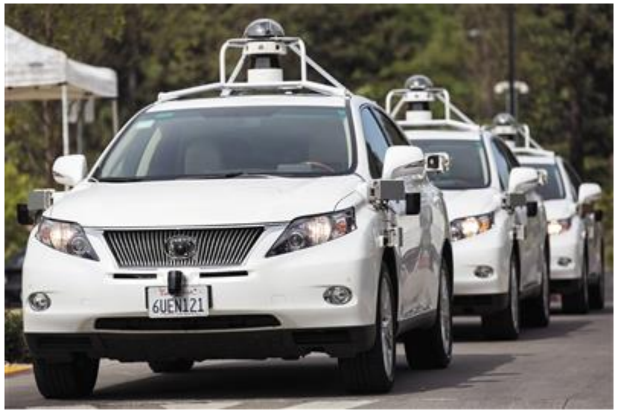
# 实时摄像机跟踪

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。

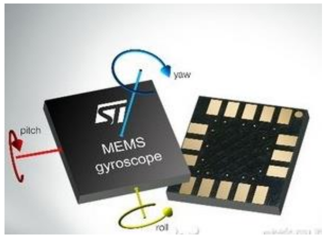
激光雷达

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

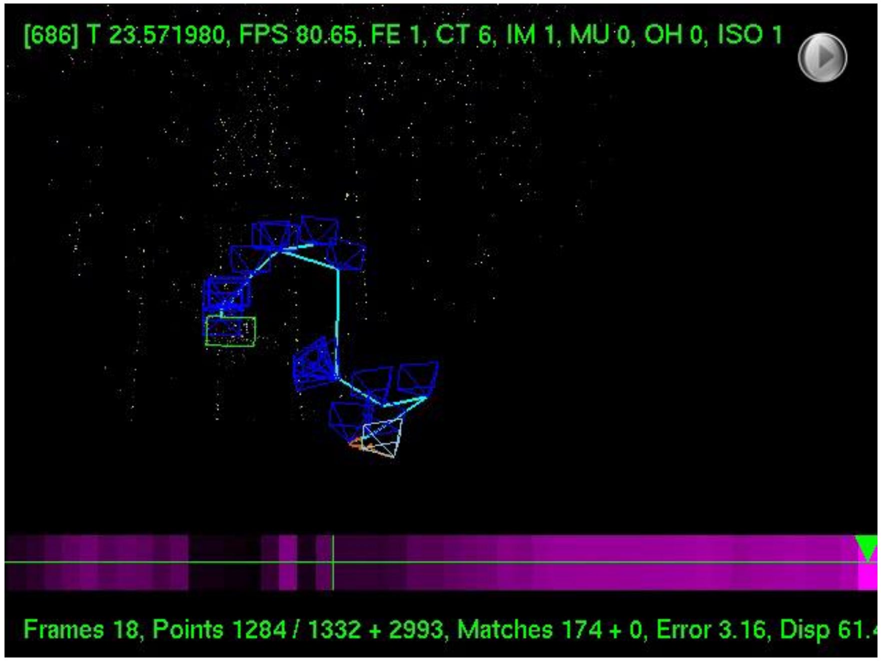
微软Kinect彩色-深度（RGBD）传感器



手机上的惯性传感器（IMU）

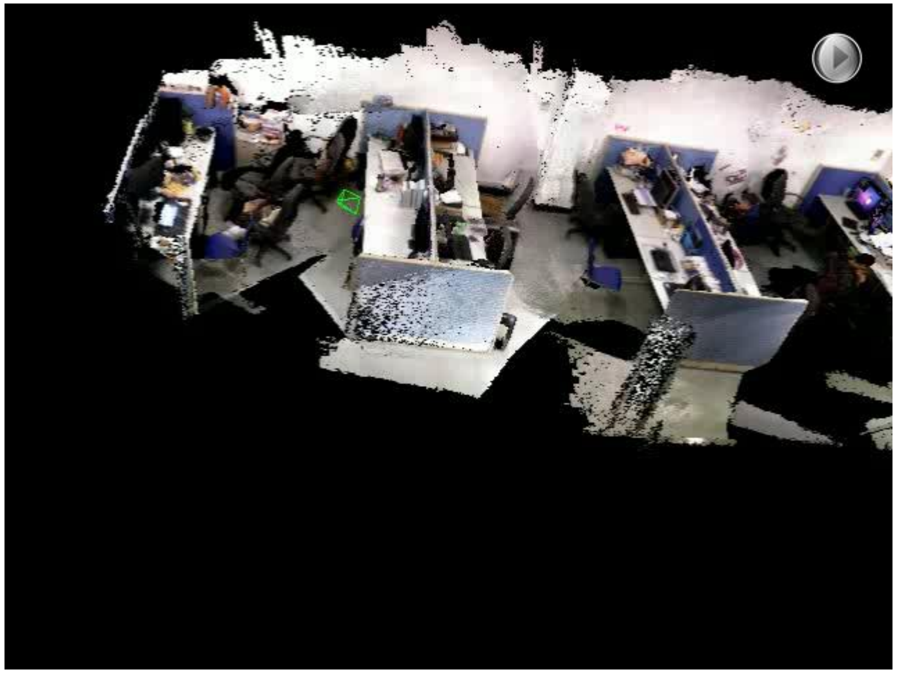
# SLAM的运行结果

- 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）



```
[686] T 23.571980, FPS 80.65, FE 1, CT 6, IM 1, MU 0, OH 0, ISO 1
Frames 18, Points 1284 / 1332 + 2993, Matches 174 + 0, Error 3.16, Disp 61.
```

稀疏SLAM



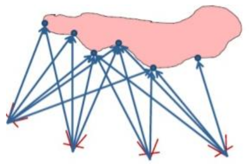
稠密SLAM

# 视觉SLAM系统常用的框架

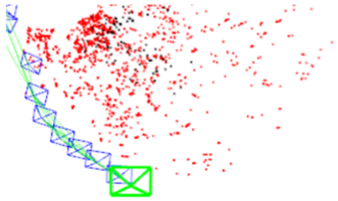
## 前台线程

视频流

输入传感器数据

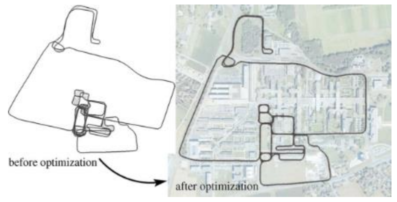


- SLAM初始化
- 特征跟踪与位姿实时求解

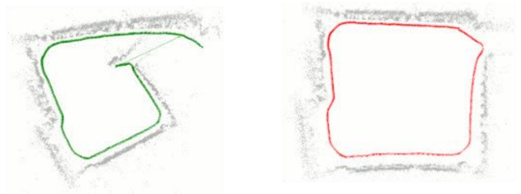


输出设备实时位姿和三维点云

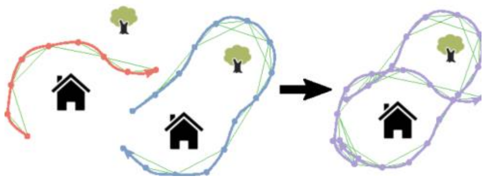
## 后台线程



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

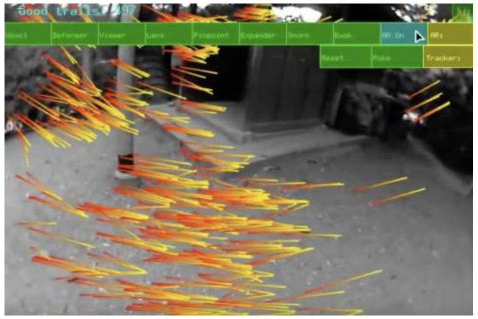
- SfM
  - 有时需要处理相机内参未知的情况，采用自定标等技术来估计出相机内参
- SLAM
  - 相机内参通常会预先标定好，固定不变
  - 单目SLAM初始化与SfM在相机内参已知情况下较类似（如Nistér的五点法[Nistér，2004]来做初始化）
  - 双目或者多目SLAM的初始化较为简单

## 单目SLAM初始化

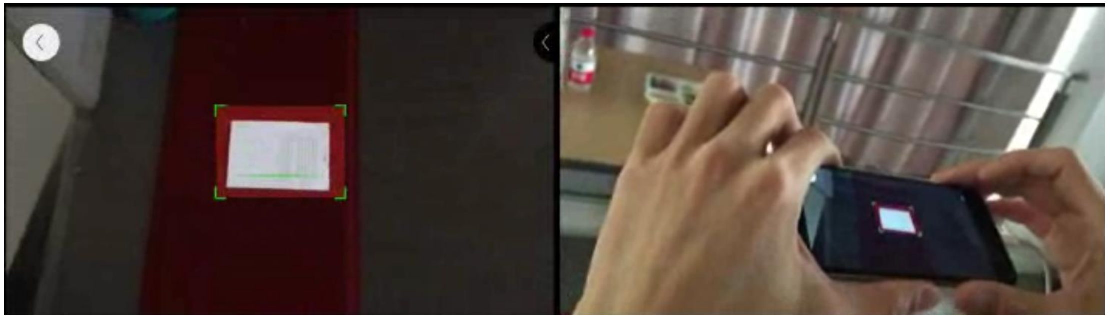
- 需求解的三维点和相机位姿相互依赖
  - 三角化三维点：需要相对位姿
  - 相对位姿：需要场景三维点
- 流程
  - 根据两帧间2D-2D的特征匹配估计相机初始位姿
  - 三角化出匹配的特征点的三维位置
  - 使用3D-2D的方法对这些特征点进行跟踪匹配，并求解相机运动

# 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

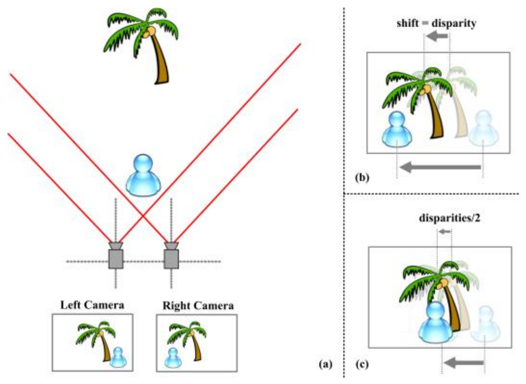
# 多目SLAM初始化

## 优势

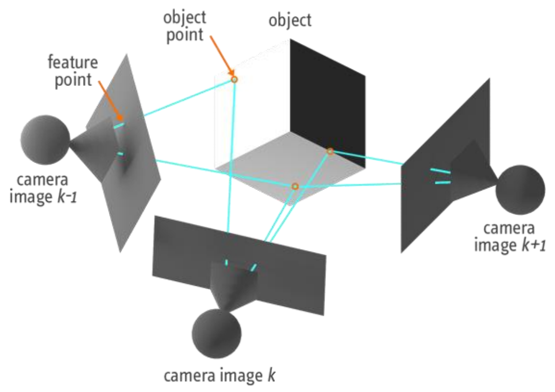
- 预先标定多个同步相机之间的相对位姿变换
- 三维信息可以通过多相机视图进行特征匹配并三角化来获得
- 相机运动直接通过3D-2D的方法求解

## 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



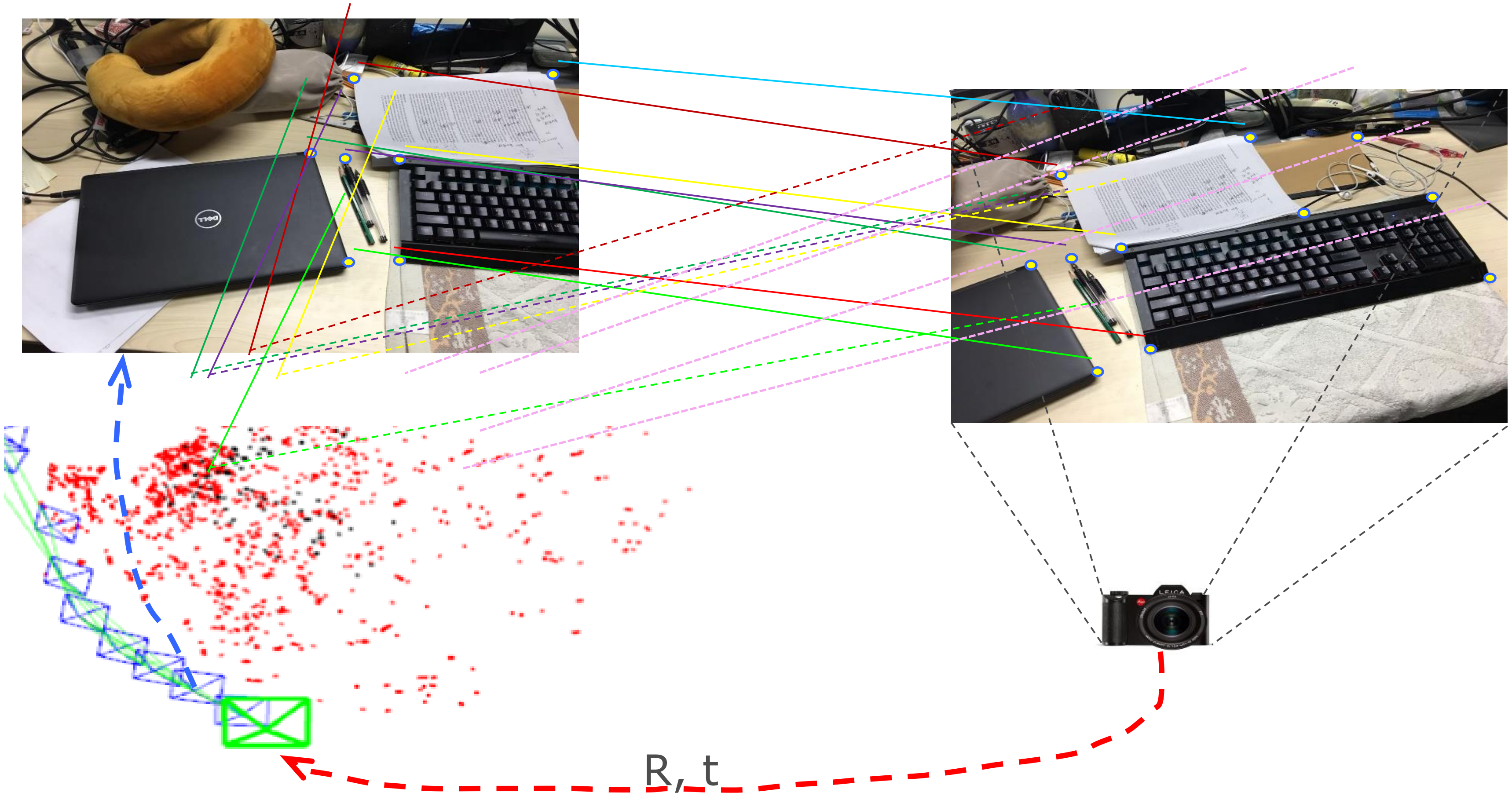
双目立体匹配



多摄像头系统特征匹配

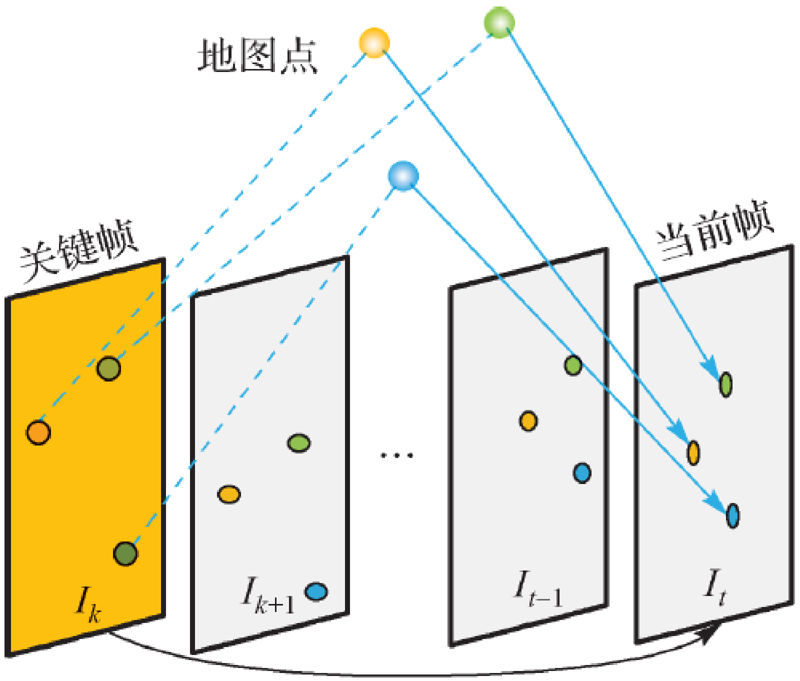
## 前台跟踪

特征检测 → 特征匹配 → 3D-2D运动估计 → 地图扩展



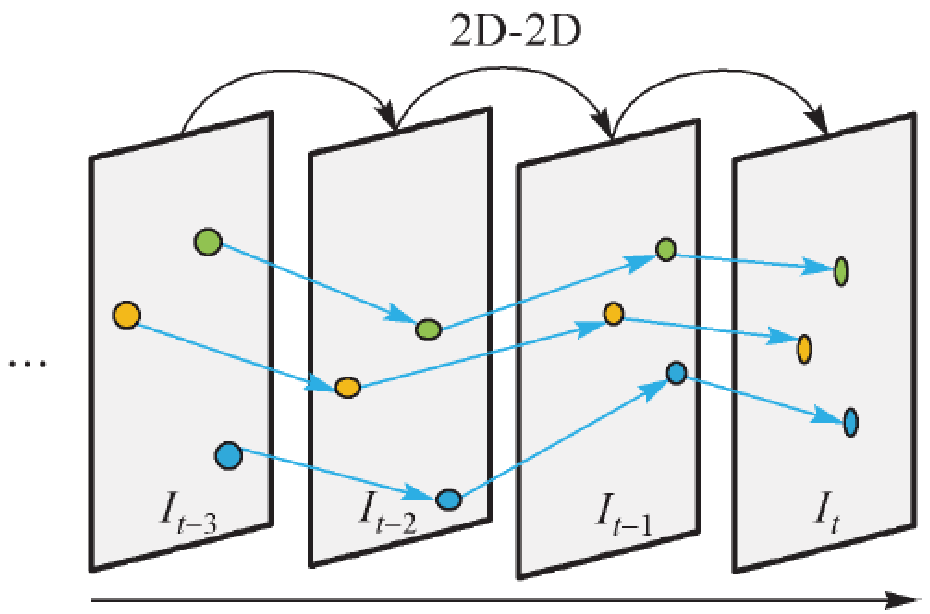
## 前台跟踪

- 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

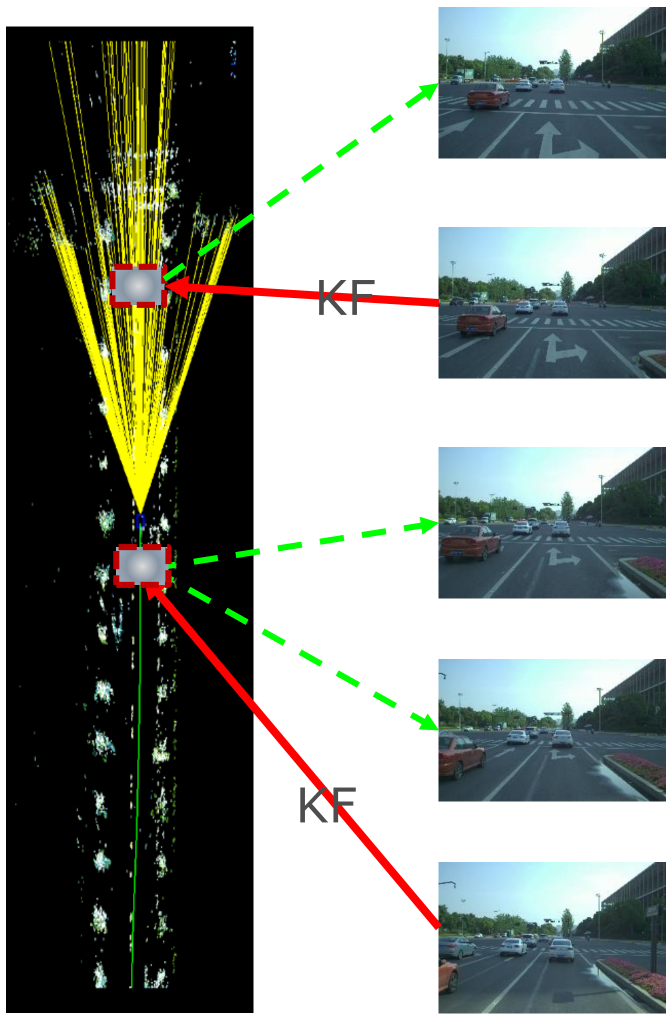


## 前台跟踪

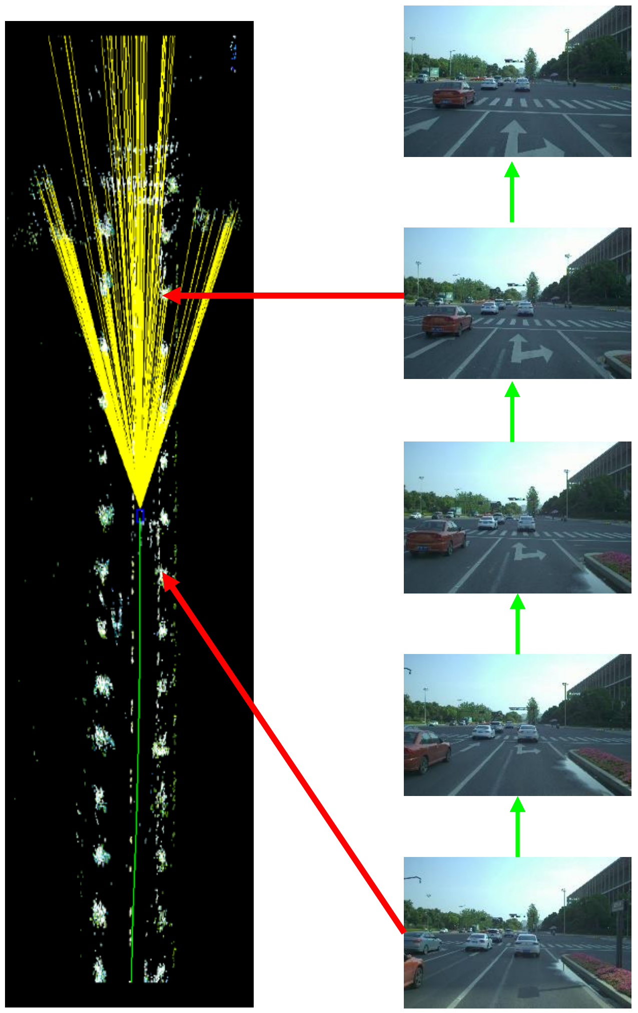
- 连续帧跟踪
  - 特征匹配基于光流法（光度误差），而非描述符匹配
  - 特征在时空上需要保持连续性，如果发生中断则认为是新的特征
  - 三维点的更新频率更高，需要能够快速完成三角化
  - 运动较快时容易LOST，容易产生累积误差



## 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
- 实时性要求高的部分需要把运算放在前端
- 运算任务更重的放在后端
  - Bundle Adjustment优化
  - 回路闭合
  - 重定位
  - 稠密三维恢复

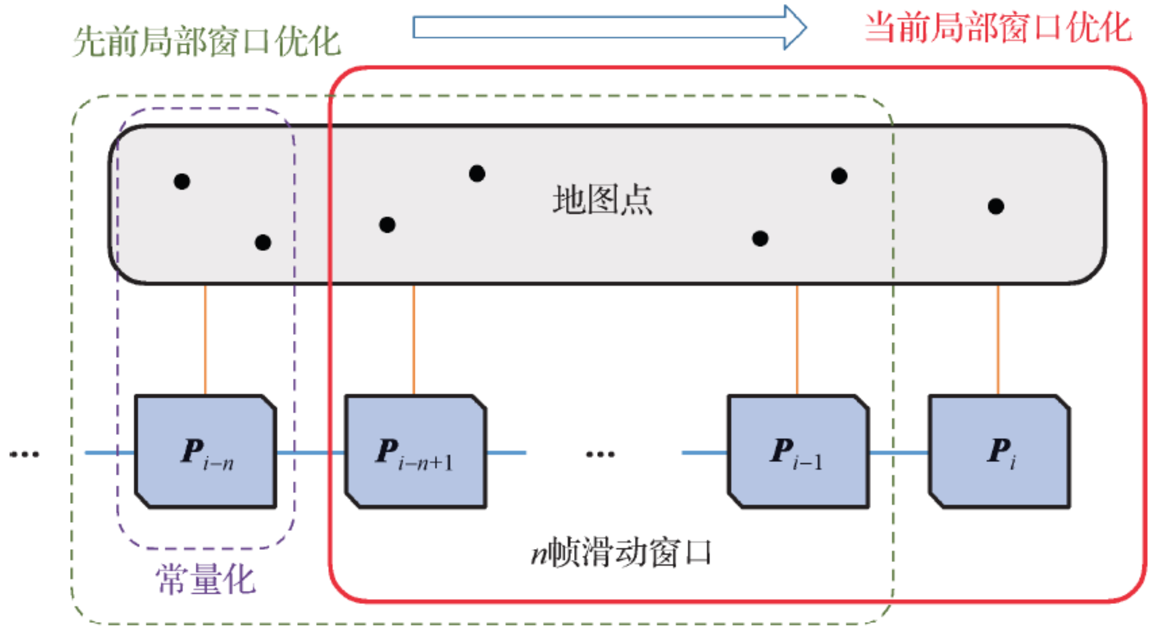
## 后台跟踪

- 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 $\mathcal{I} \triangleq I_t, I_{t+1}, I_{t+2}, \cdots, I_{t+k}$ 最小化重投影误差

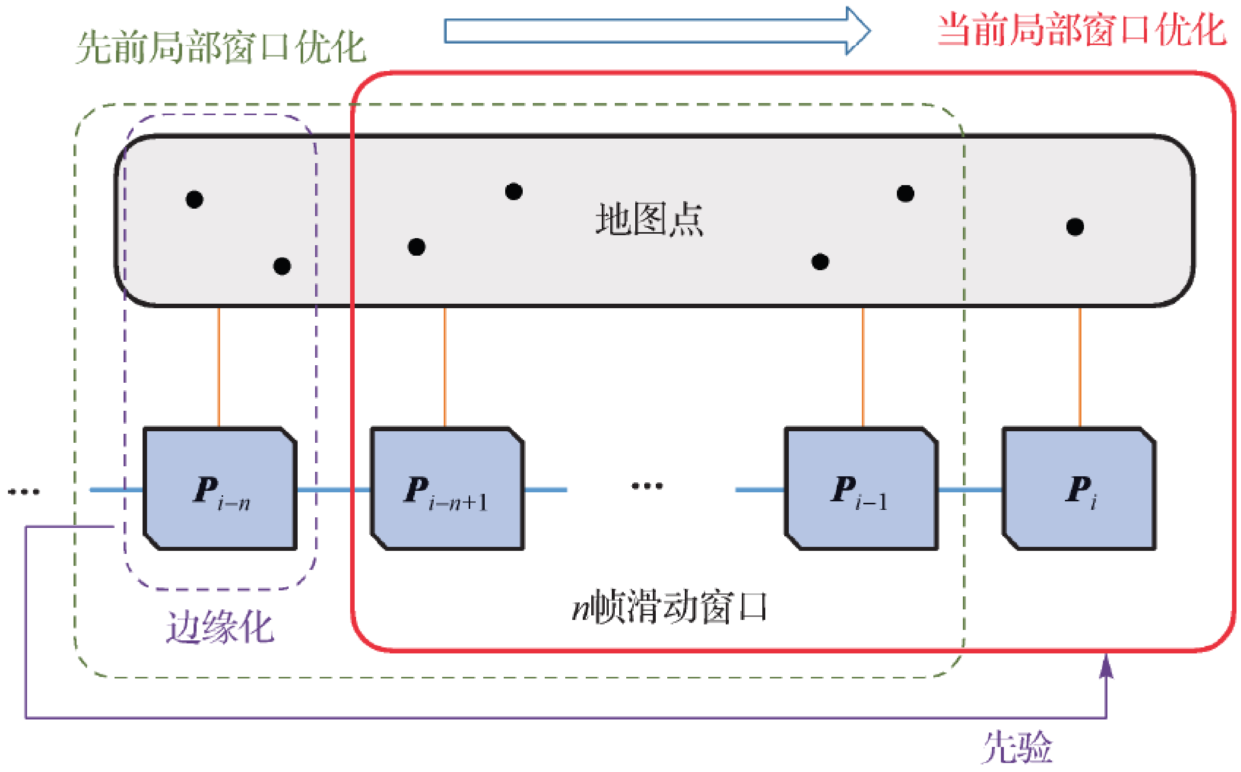
$$\arg\min_{\mathcal{I}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathcal{I}_i} \left\| \Pi\left(\mathrm{K}_i\left(\mathrm{R}_i \mathbf{p}_j + \mathbf{t}_i\right)\right) - \mathbf{x}_{ij} \right\|_{\Sigma_{ij}}^2$$



# 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

$$\arg\min_{C,M}\left( \| r_{\mathcal{I}} \|^2_{\Sigma_{\mathcal{I}}} + \sum_{k=i-n+1}^{i} \sum_{x_{kj}\in I_k} \| \pi(K_k(R_k X_j + t_k)) - x_{kj} \|^2_{\Sigma_{kj}} \right)$$

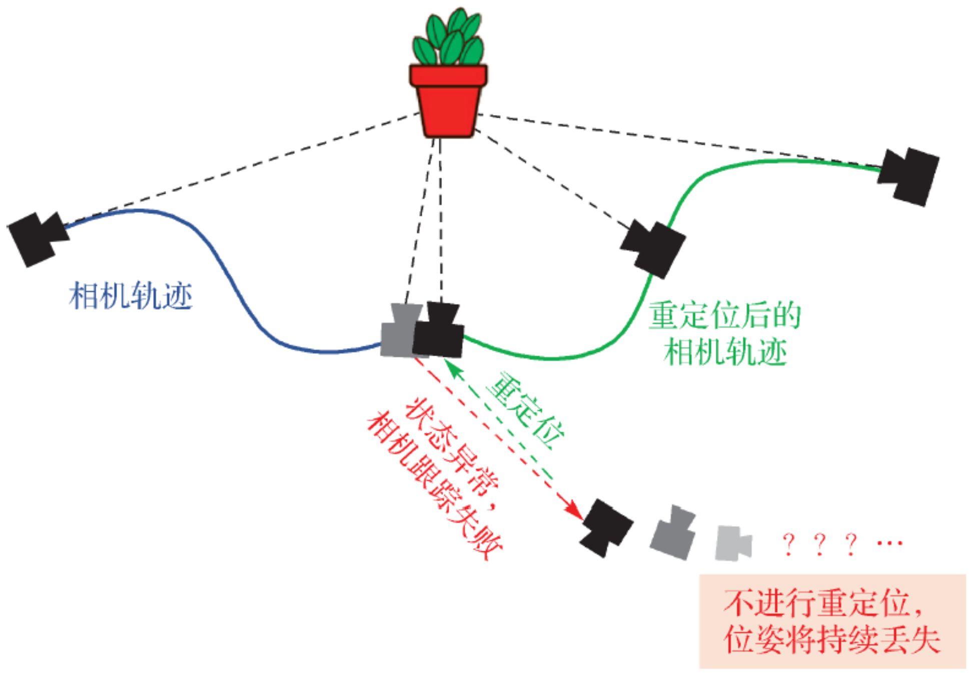


## 后台跟踪

- 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，EIBA、ICE-BA）
  - 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

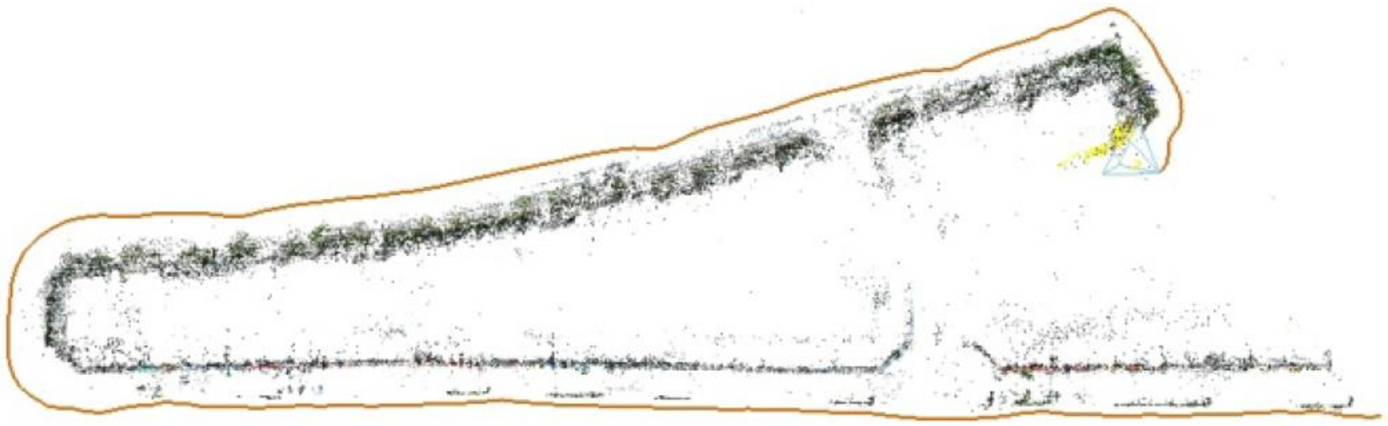
# 重定位

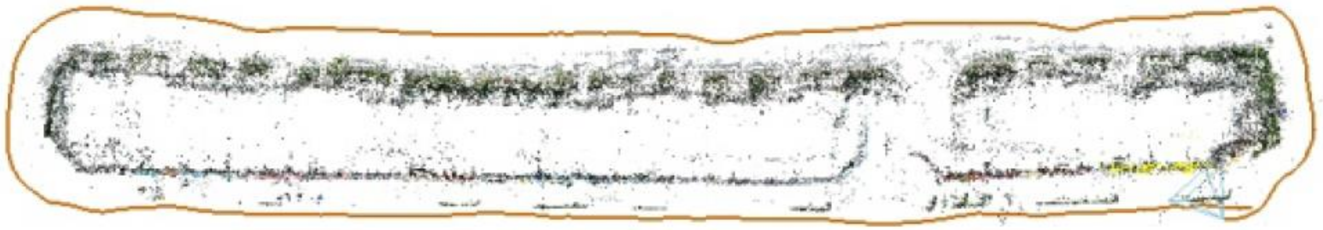
- 必要性
  - 视觉 SLAM有时会出现跟踪失败的情况，图像质量过差或者图像内容缺少特征都可能导致一段时间内跟踪失败。
  - 需要从相机跟踪失败的状态恢复到正常跟踪状态。
- 目标
  - 在连续跟踪失败的情况下快速重新得到当前的精确位姿



# 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

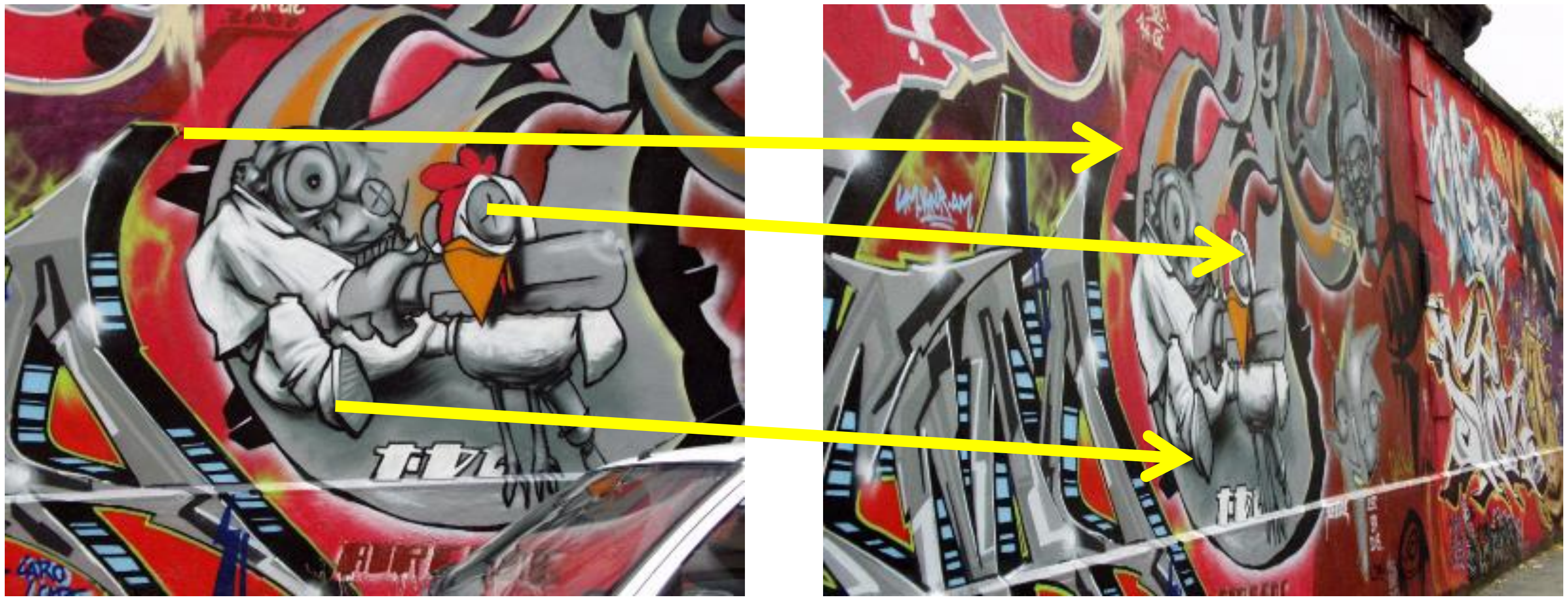
- 相似性
  - 初始操作类似：寻找当前场景与已经生成的地图的联系，找到曾经访问过的场景。
  - 本质上都是一个图像检索的过程。
- 优化目标不同
  - 重定位只需得到相机的当前位姿
  - 回路闭合需要修正整个回路的轨迹以及相关的三维点坐标

## 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

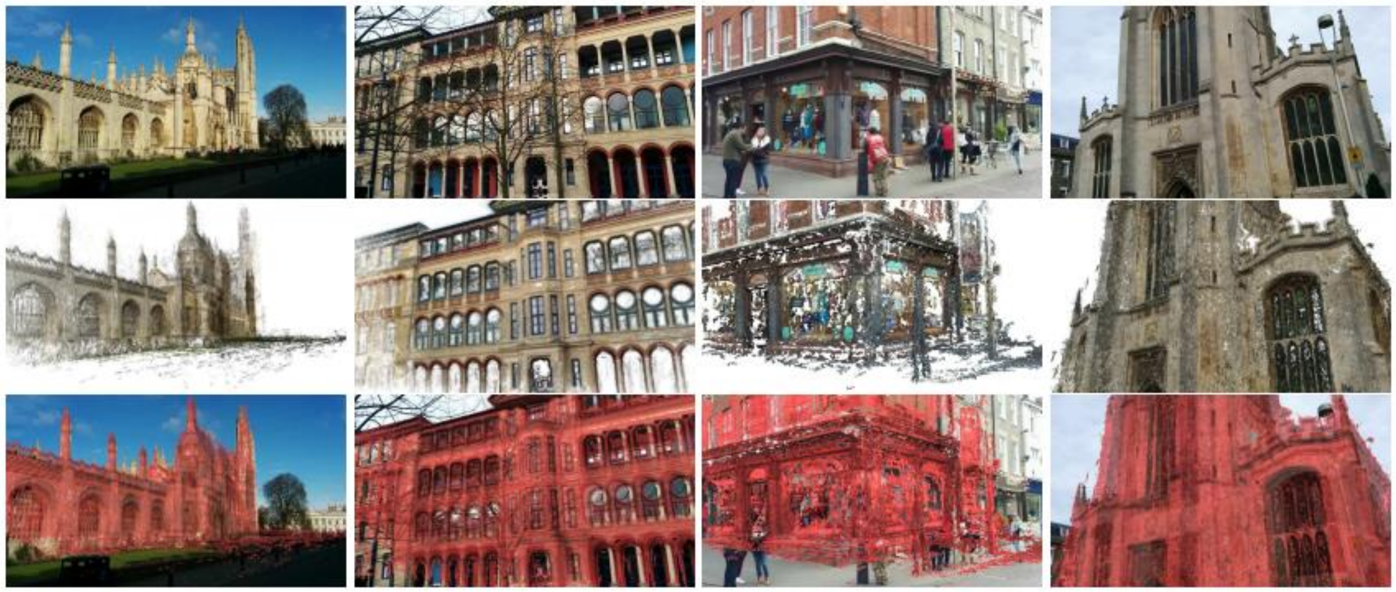
- 基于各种局部特征点的检索方法
  - 可以在一定程度上容忍视角的变化
  - 对重复纹理和图像模糊的容忍度较差



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

$$x_k \sim N(\underbrace{\hat{x}_k}_{\text{mean}}, \underbrace{P_k}_{\text{covariance}})$$

- 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

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

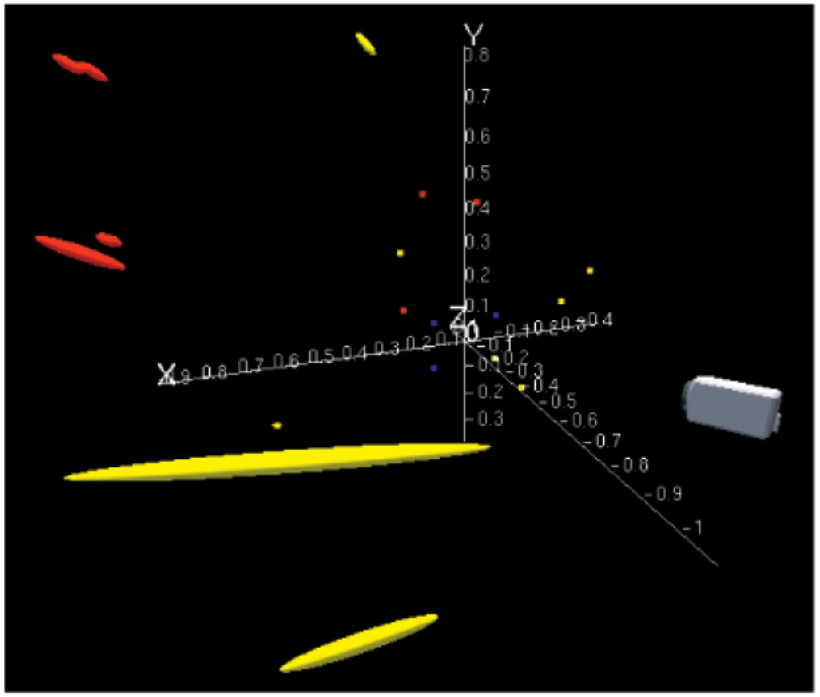
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - h(\hat{x}_{k|k-1}))$$

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

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

# MonoSLAM

- Map representation



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

camera state

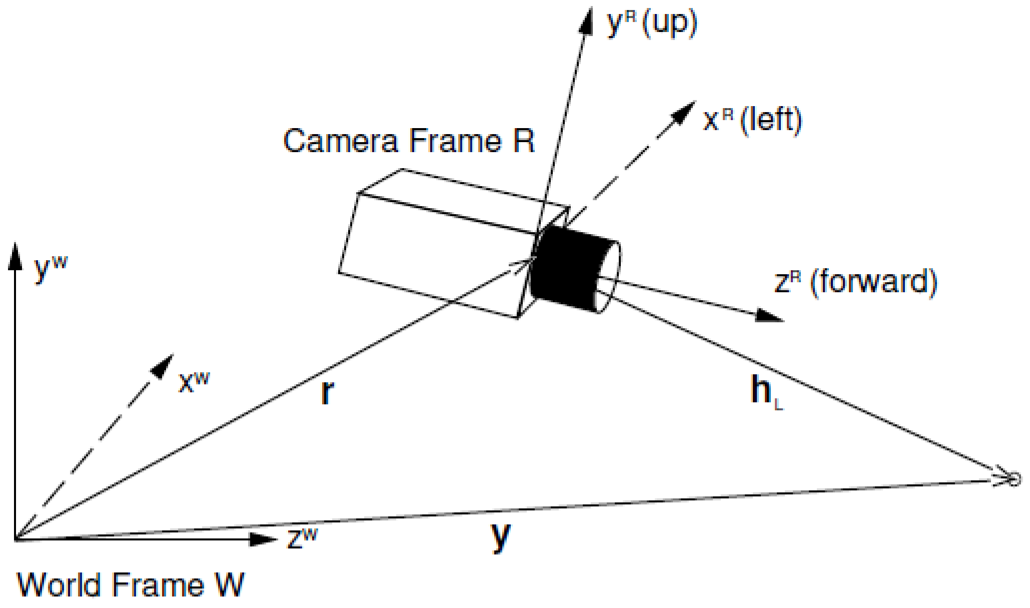
point state

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

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

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

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

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

## MonoSLAM

- Predicted features position

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

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

- Innovation covariance
  - Elliptical feature search region

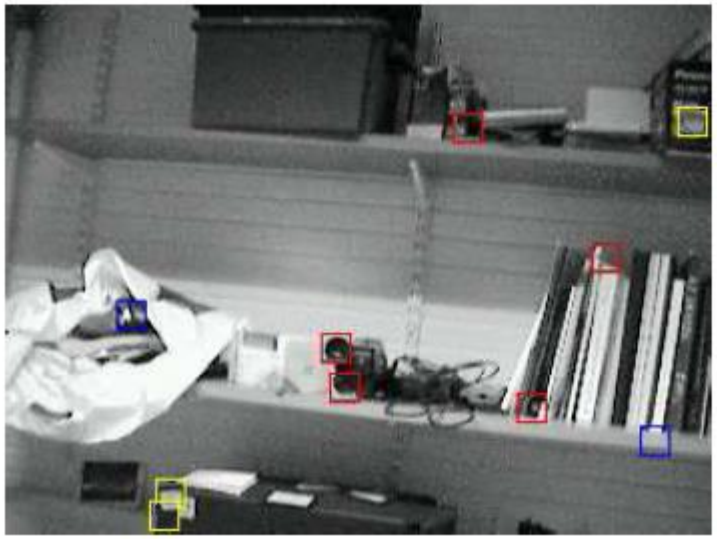
$$S_i = J_C P_{CC} J_C^T + J_C P_{CX_i} J_{X_i}^T + J_{X_i} P_{X_i C} J_C^T + J_{X_i} P_{X_i X_i} J_{X_i}^T + R$$

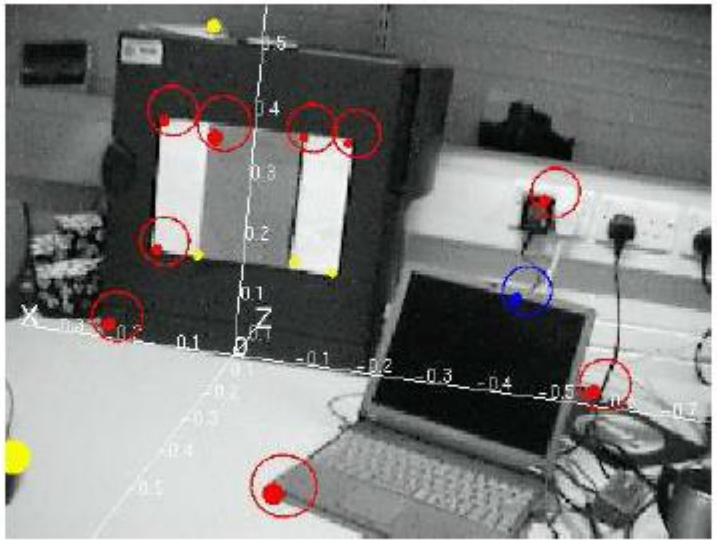
$$J_C = \frac{\partial z_i}{\partial C}$$

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

## MonoSLAM

- Active search


Shi and Tomasi Feature

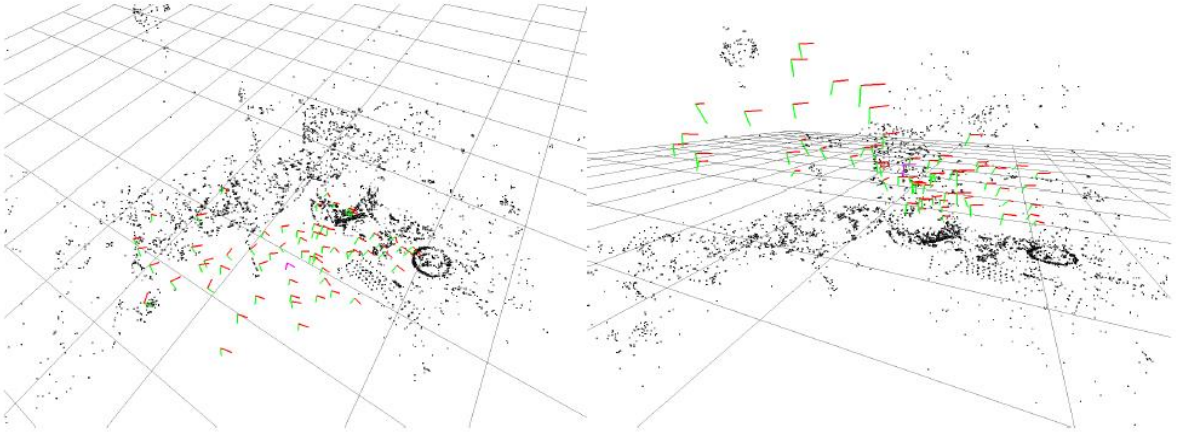

Elliptical search region

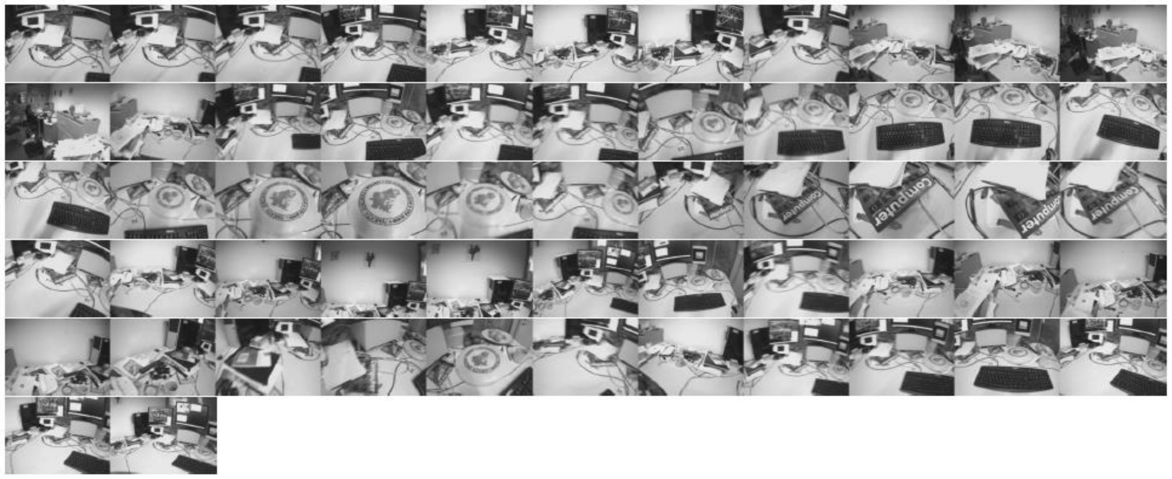
## MonoSLAM

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

## PTAM: Parallel Tracking and Mapping

- Map representation

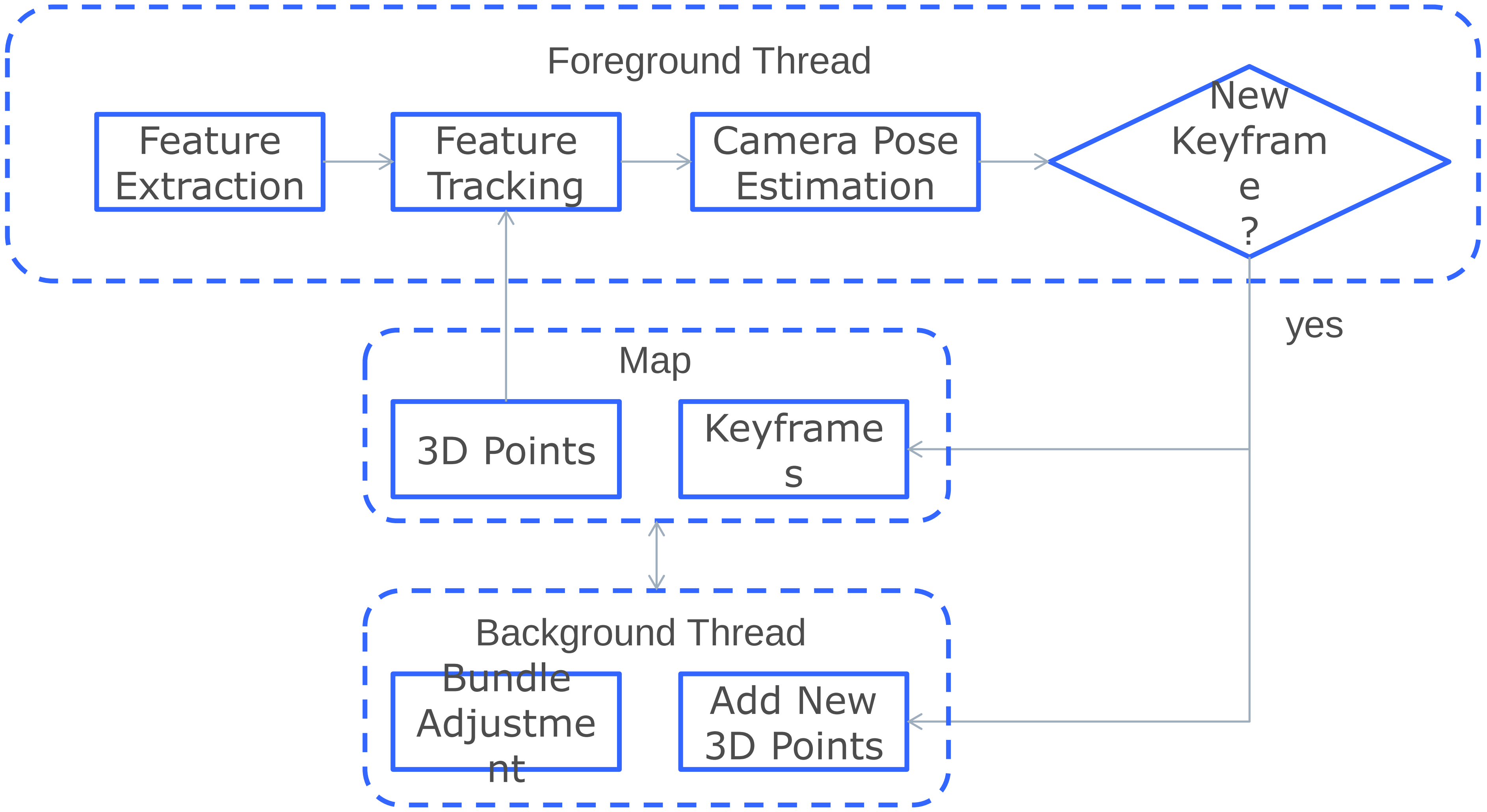




G. Klein and D. W. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR), 2007.

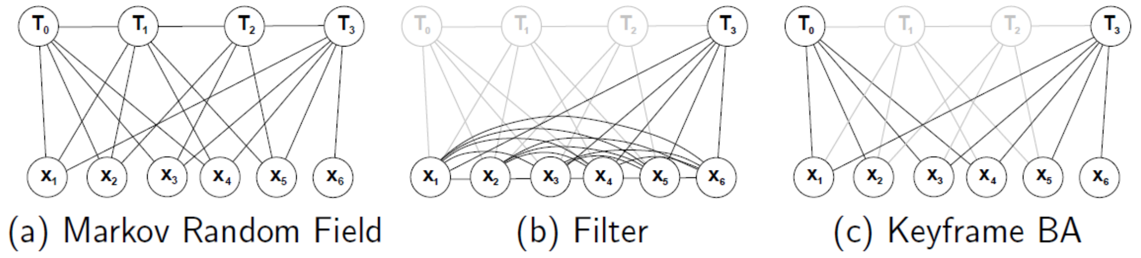
# PTAM: Parallel Tracking and Mapping

- Overview

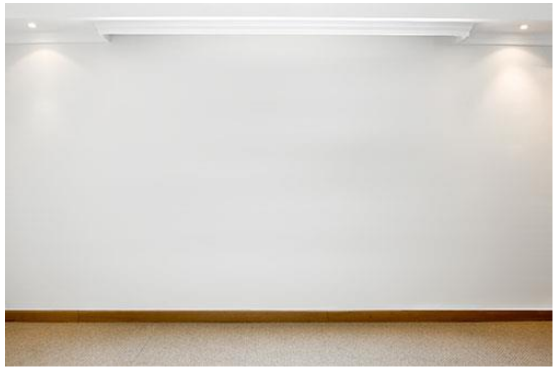


# Keyframe-based SLAM vs Filtering-based SLAM

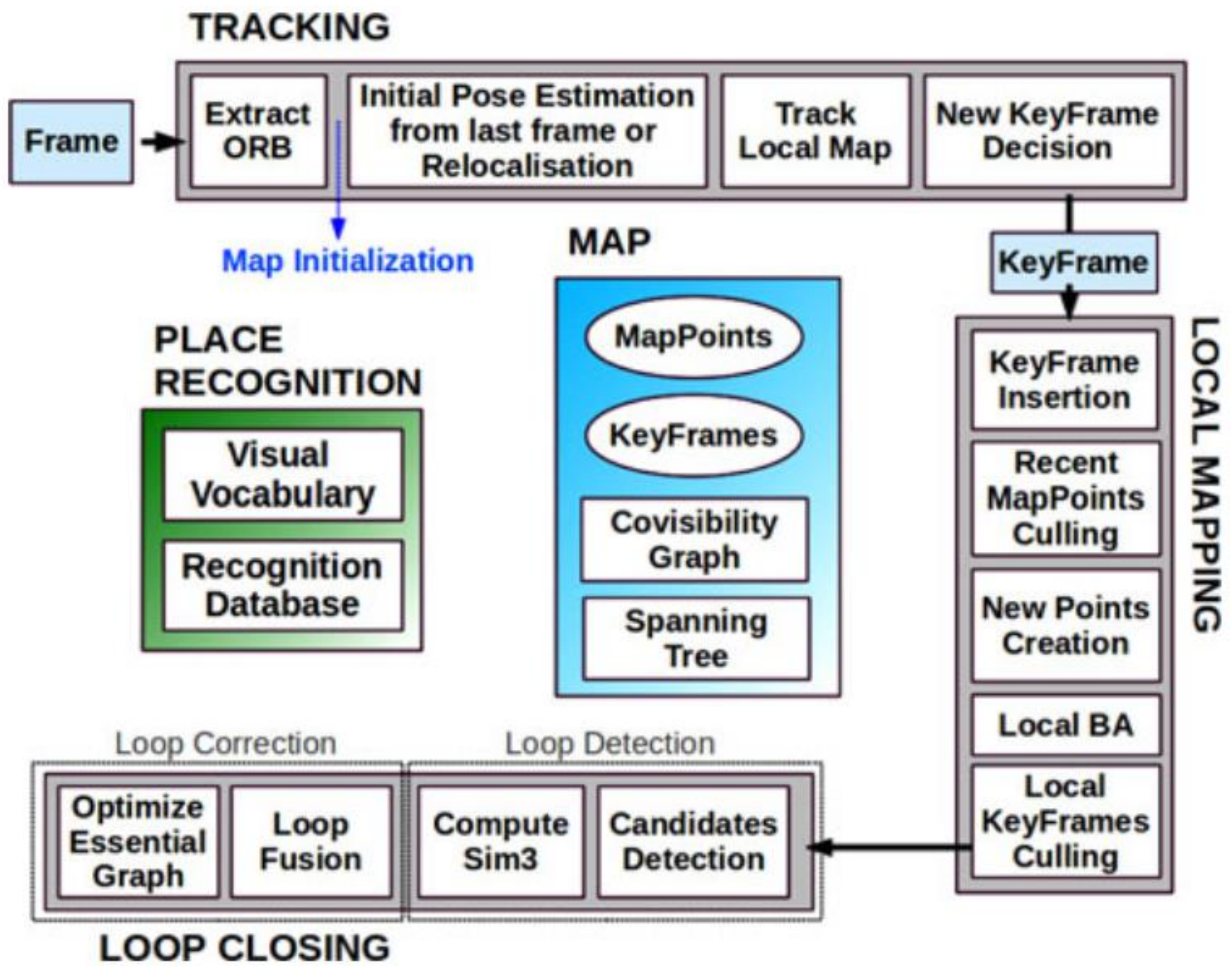
- Advantages
  - Accuracy
  - Efficiency
  - Scalability
- Disadvantages
  - Sensitive to strong rotation
- Challenges for both
  - Fast motion
  - Motion blur
  - Insufficient texture



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



# ORB-SLAM

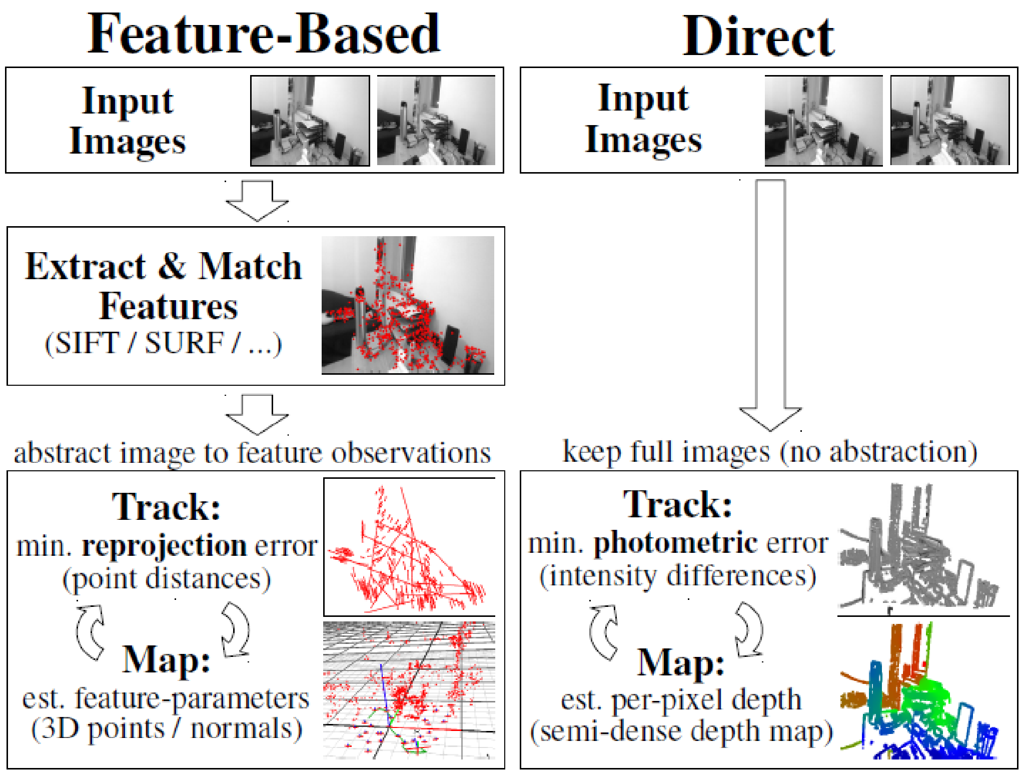


Raul Mur-Artal, J. M. M. Montiel, Juan D. Tardós: ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Trans. Robotics 31(5): 1147-1163 (2015).

## 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.

## Direct Tracking



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

## Direct Tracking

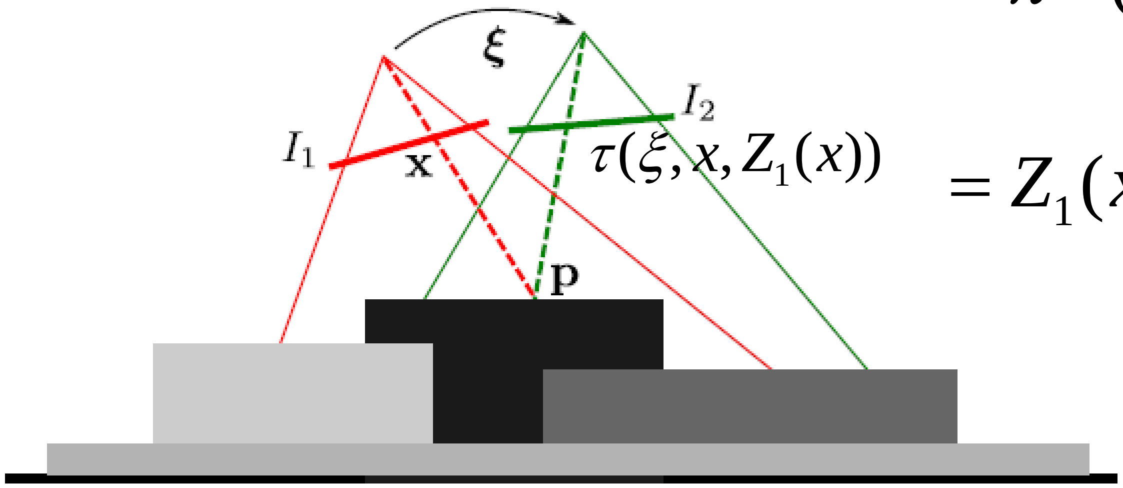
- Goal
  - Estimate the camera motion $\xi$ by aligning intensity images $I_1$ and $I_2$ with depth map $Z_1$ of $I_1$
- Assumption

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

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

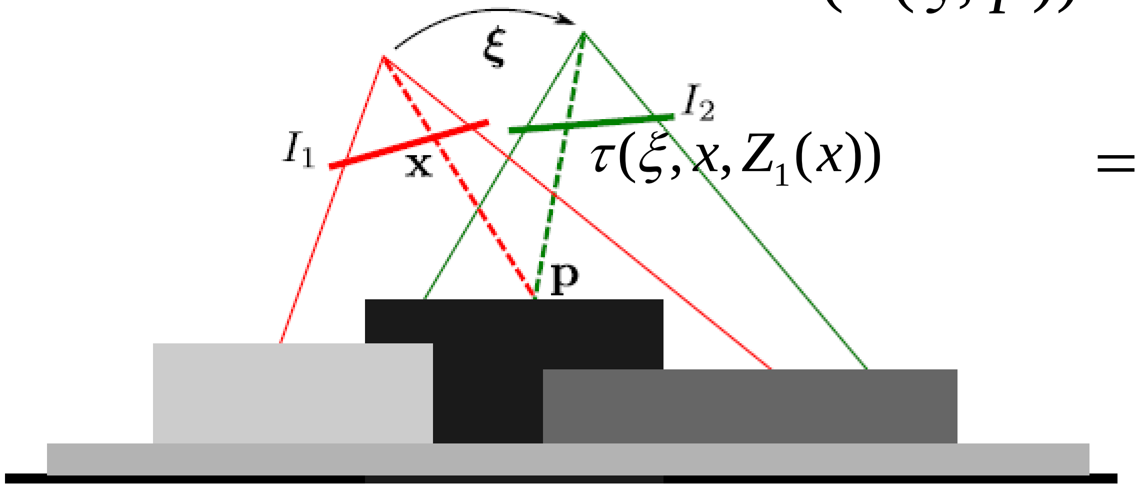
# Direct Tracking

- Warping function

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

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

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

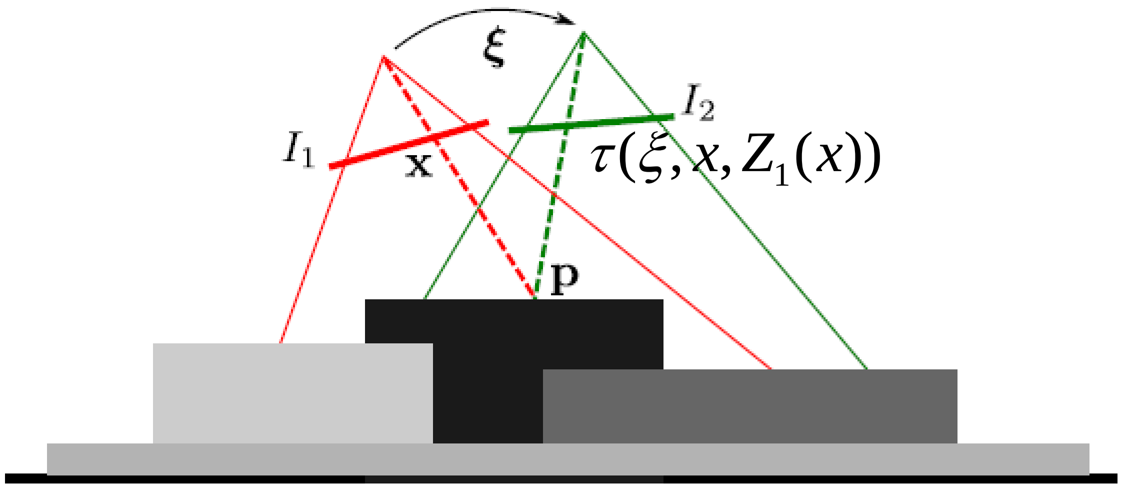


Christian Kerl, Jürgen Sturm, Daniel Cremers: Robust odometry estimation for RGB-D cameras. ICRA 2013: 3748-3754

## 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

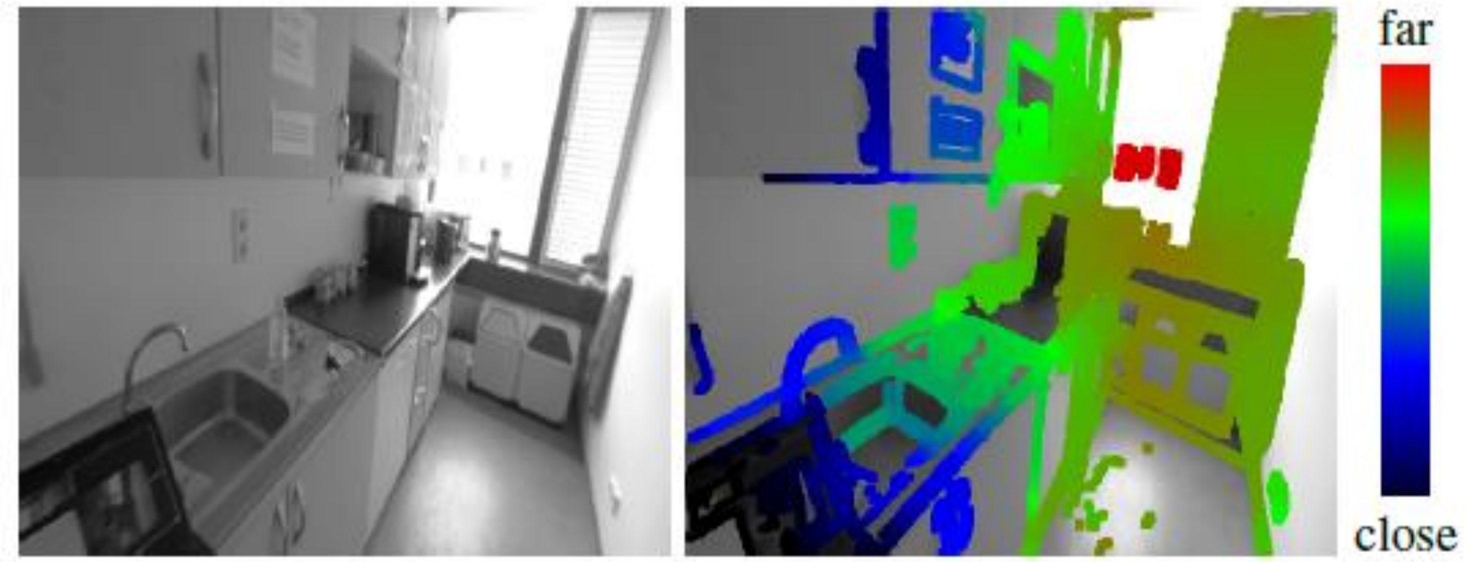
- Residual of the *k*-th pixel

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

- 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

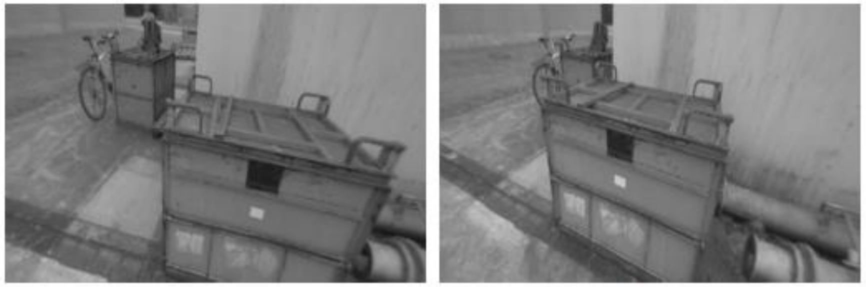
- Keyframe representation

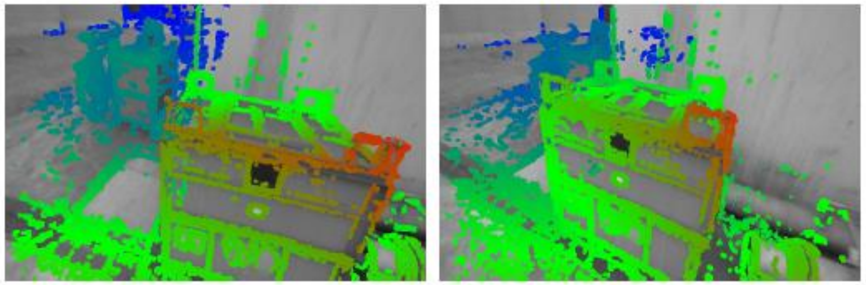
$$K_i = (I_i, D_i, V_i)$$

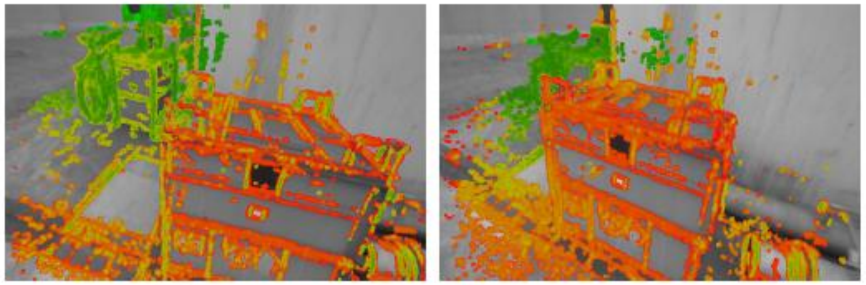
$$i_i = I_i(x)$$ image intensity

$$d_i = D_i(x)$$ inverse depth

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

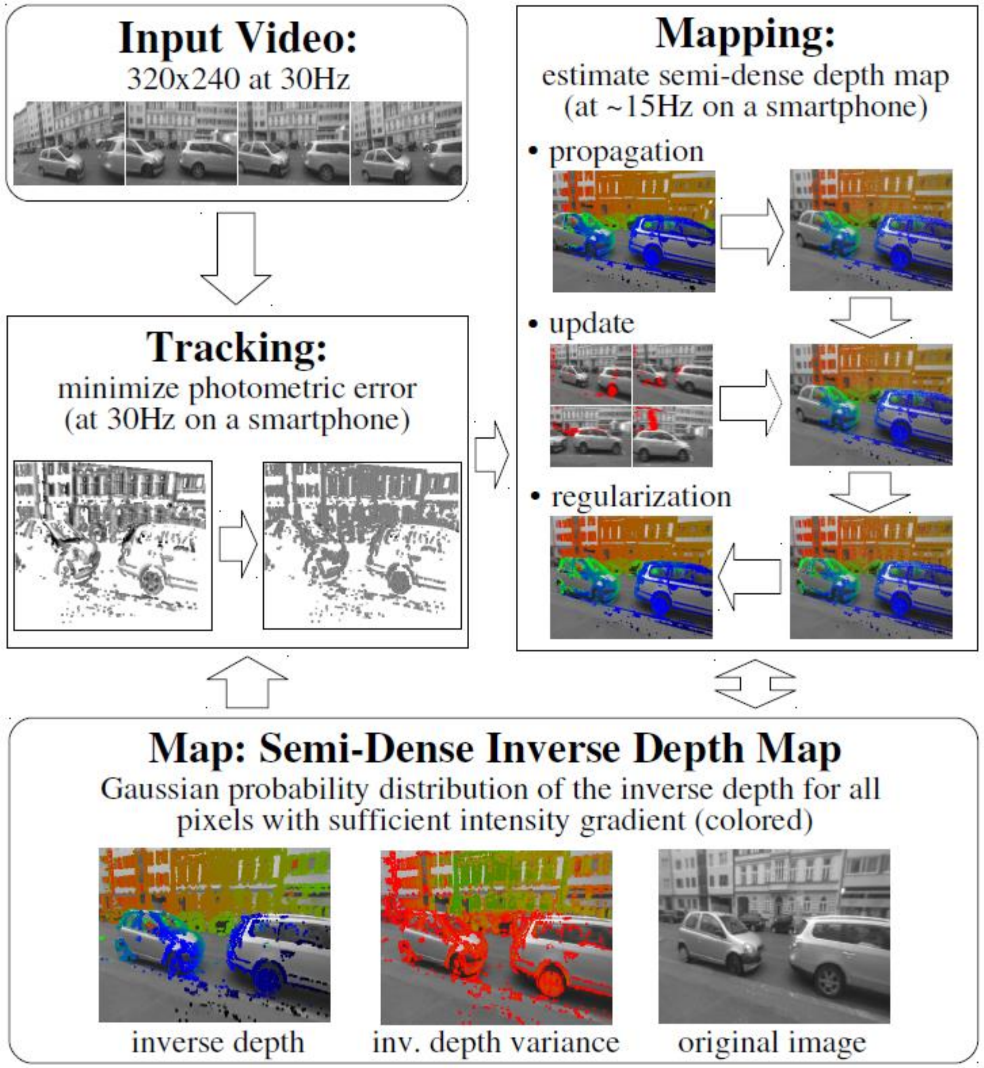

(a) camera images $I$


(b) estimated inverse depth maps $D$


(c) inverse depth variance $V$

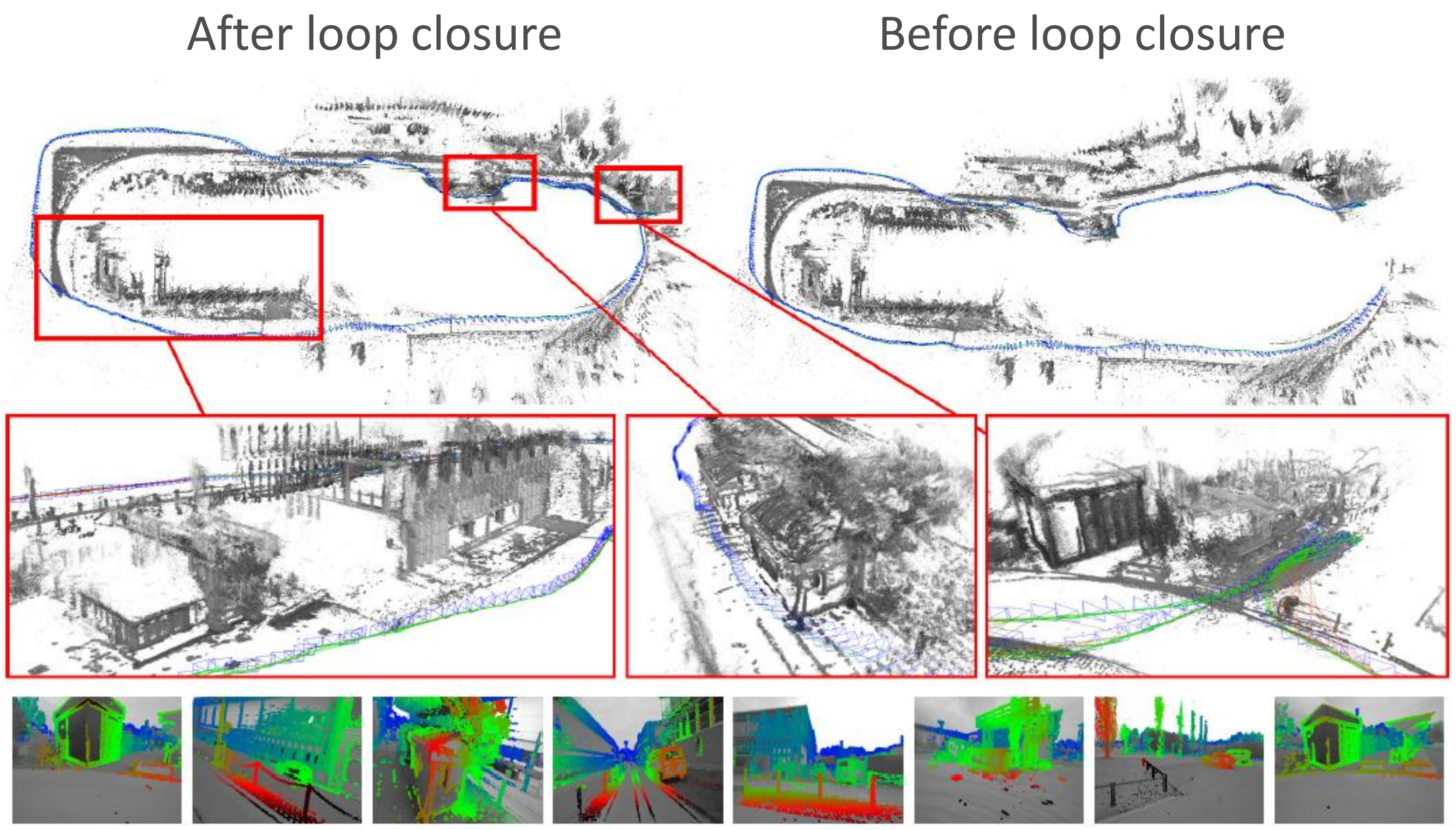
# Semi-Dense Visual Odometry

- Overview



Input Video:
320x240 at 30Hz
Tracking:
minimize photometric error
(at 30Hz on a smartphone)
Mapping:
estimate semi-dense depth map
(at ~15Hz on a smartphone)
• propagation
• update
• regularization
Map: Semi-Dense Inverse Depth Map
Gaussian probability distribution of the inverse depth for all pixels with sufficient intensity gradient (colored)
inverse depth
inv. depth variance
original image

# LSD-SLAM



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

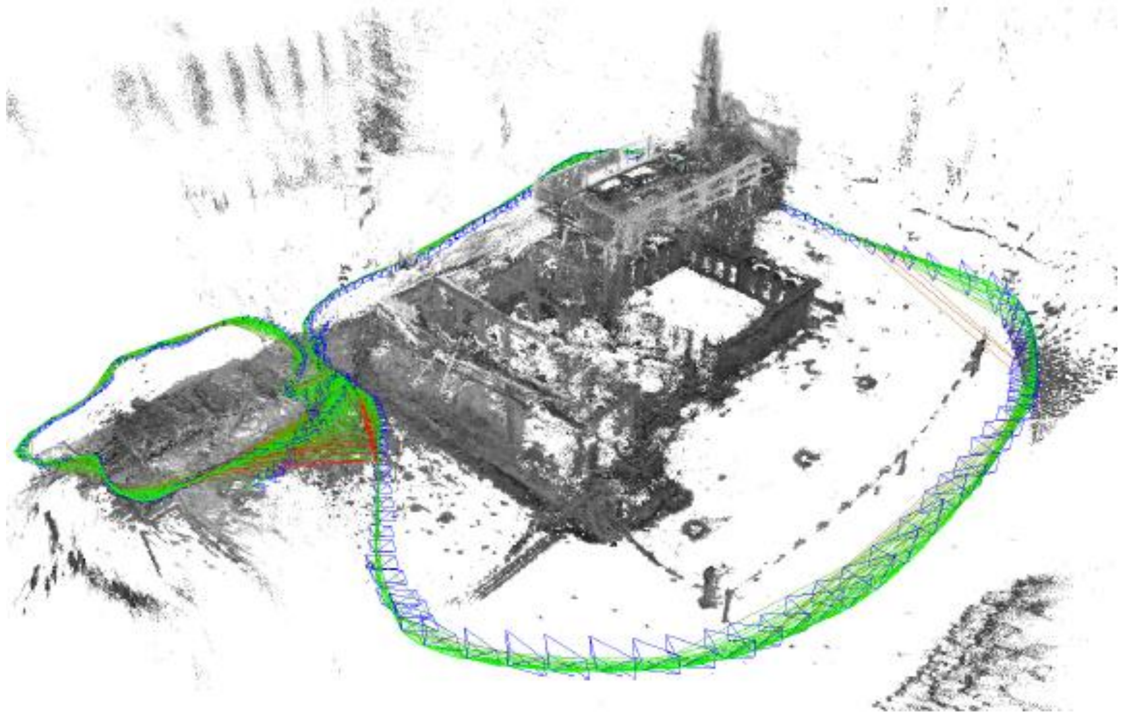
# LSD-SLAM

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

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

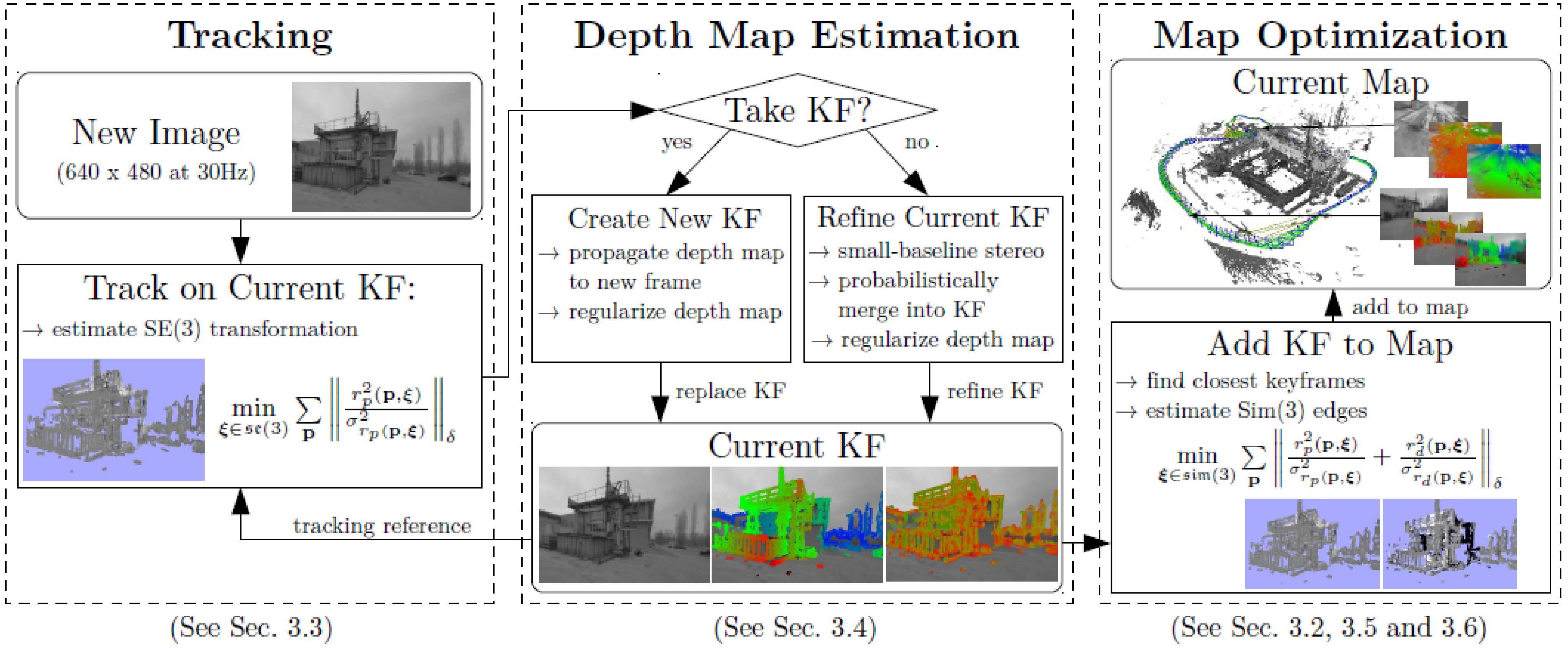
  - Edge: similarity transformation

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



# LSD-SLAM

- Overview



Tracking
New Image
(640 x 480 at 30Hz)
Track on Current KF:
→ estimate SE(3) transformation
$\min_{\xi \in se(3)} \sum_{\mathbf{p}} \left\| \frac{r_p^2(\mathbf{p},\xi)}{\sigma^2_{r_p(\mathbf{p},\xi)}} \right\|_\delta$
tracking reference
Depth Map Estimation
Take KF?
yes
no
Create New KF
→ propagate depth map to new frame
→ regularize depth map
Refine Current KF
→ small-baseline stereo
→ probabilistically merge into KF
→ regularize depth map
replace KF
refine KF
Current KF
Map Optimization
Current Map
add to map
Add KF to Map
→ find closest keyframes
→ estimate Sim(3) edges
$\min_{\xi \in sim(3)} \sum_{\mathbf{p}} \left\| \frac{r_p^2(\mathbf{p},\xi)}{\sigma^2_{r_p(\mathbf{p},\xi)}} + \frac{r_d^2(\mathbf{p},\xi)}{\sigma^2_{r_d(\mathbf{p},\xi)}} \right\|_\delta$
(See Sec. 3.3)
(See Sec. 3.4)
(See Sec. 3.2, 3.5 and 3.6)

## ENFT-SLAM: ENFT-based Large-Scale Monocular SLAM

# ENFT-SLAM

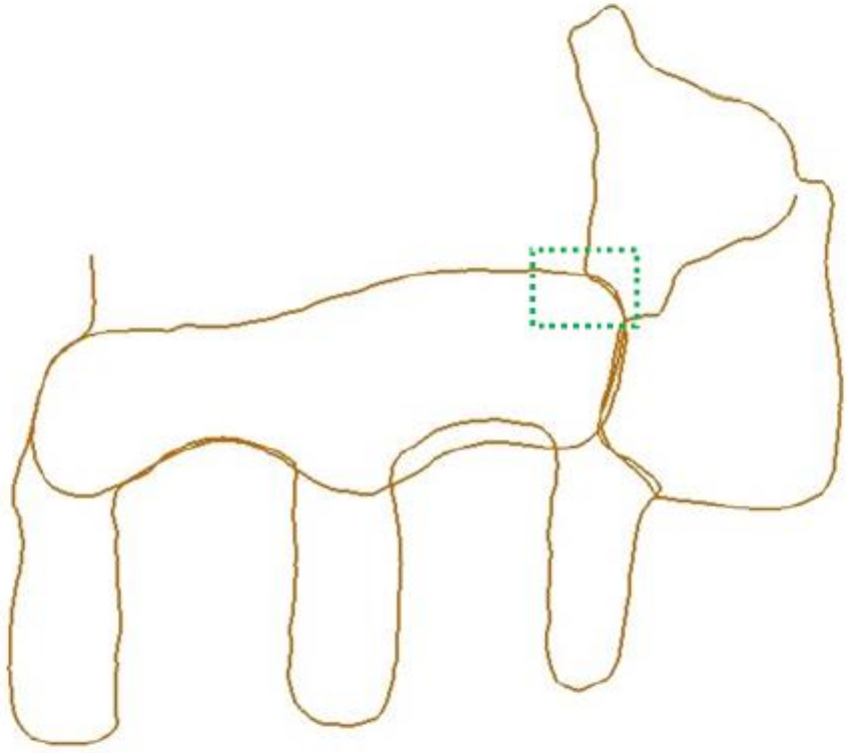
- Tracking
  - Use ENFT feature tracking
- Mapping with Loop Closure
  - Detect loops with modified non-consecutive track matching
  - Segment-based BA for loop closure

## Non-Consecutive Track Matching for Loop Detection

- Matching Matrix is intractable
  - The number of keyframes is dynamically increased.
- Matching Vector
  - Measure the overlapping confidence between current fame and keyframes.
  - Detect loops
    - Quick matching vector estimation with track descriptors.
    - Match current frame with the selected keyframes.

## RealTime SLAM for Garden Sequence

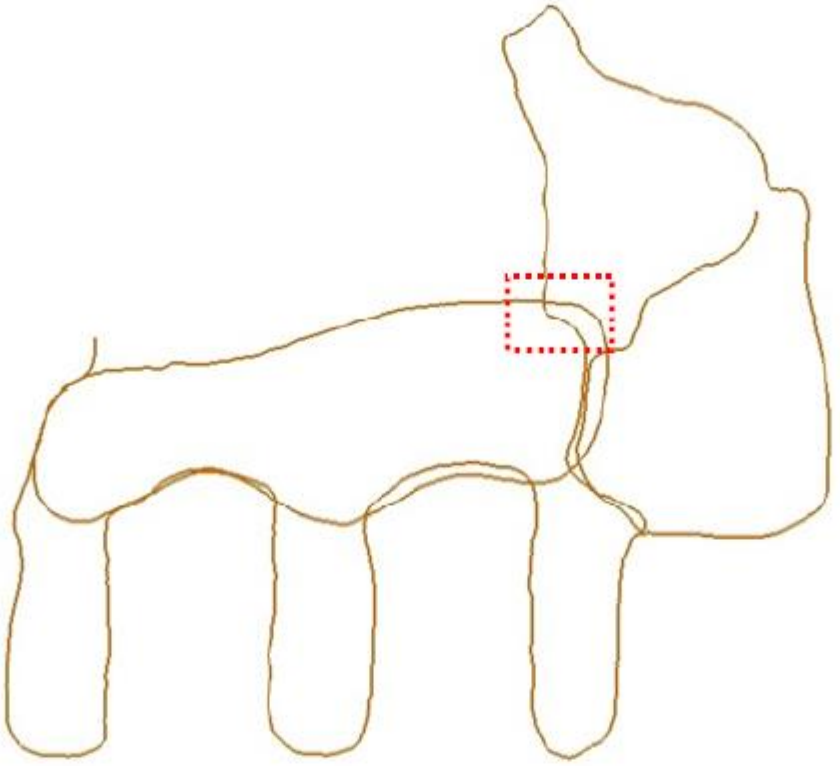
## Comparison with ORB-SLAM in Garden 01 Sequence



ENFT-SLAM

Non-consecutive Track Matching

Segment-based BA

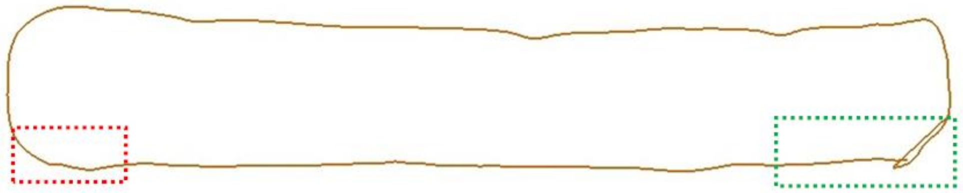


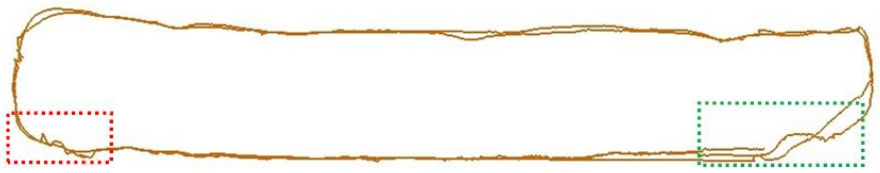
ORB-SLAM

Bag-of-words Place Recognition

Pose Graph Optimization + Traditional BA

## Comparison with ORB-SLAM in Street Sequence





ENFT-SLAM

Non-consecutive Track Matching

Segment-based BA

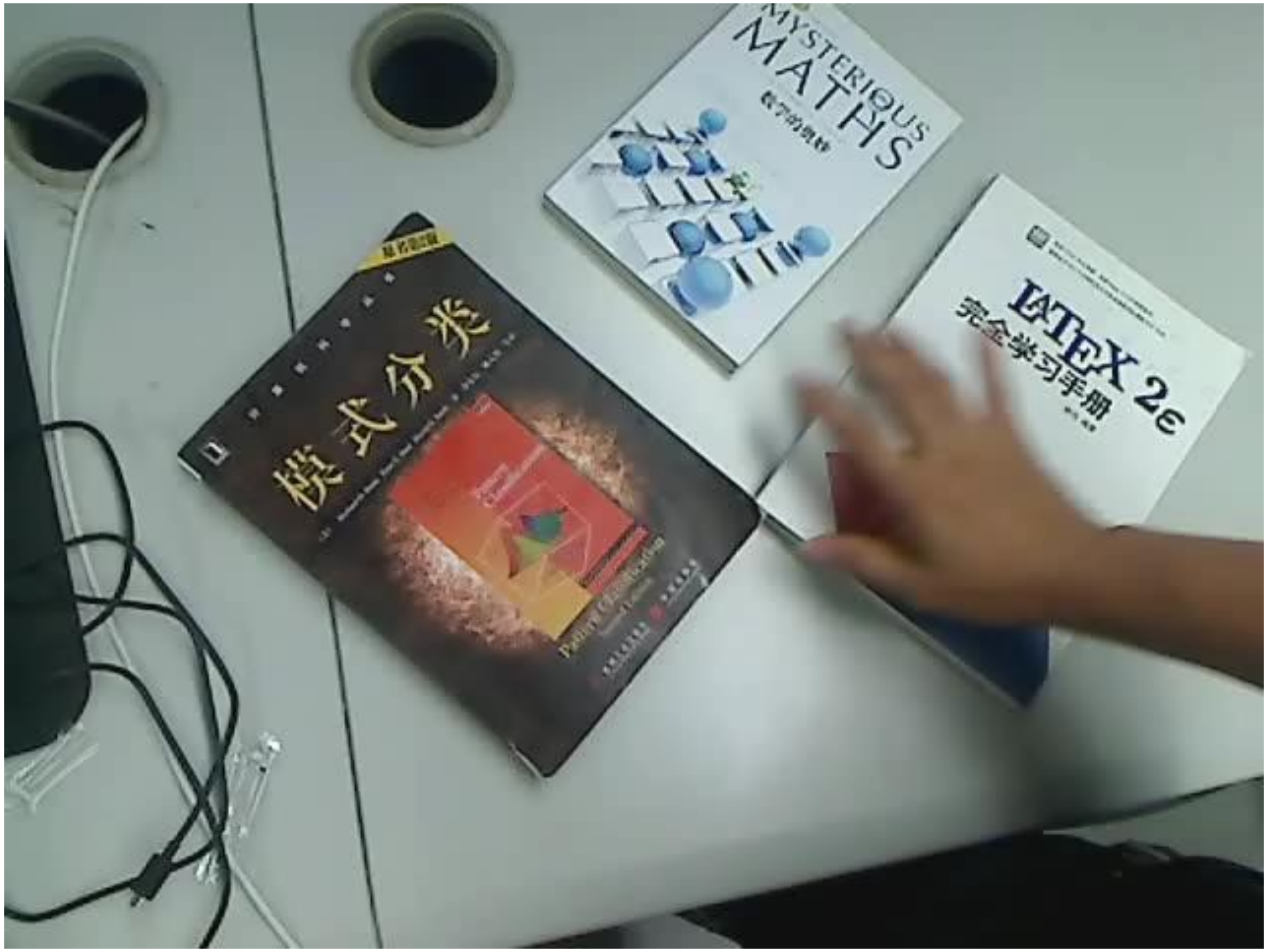
ORB-SLAM

Bag-of-words Place Recognition

Pose Graph Optimization + Traditional BA

## Key Issues for SLAM in Dynamic Environments

- Gradually changing



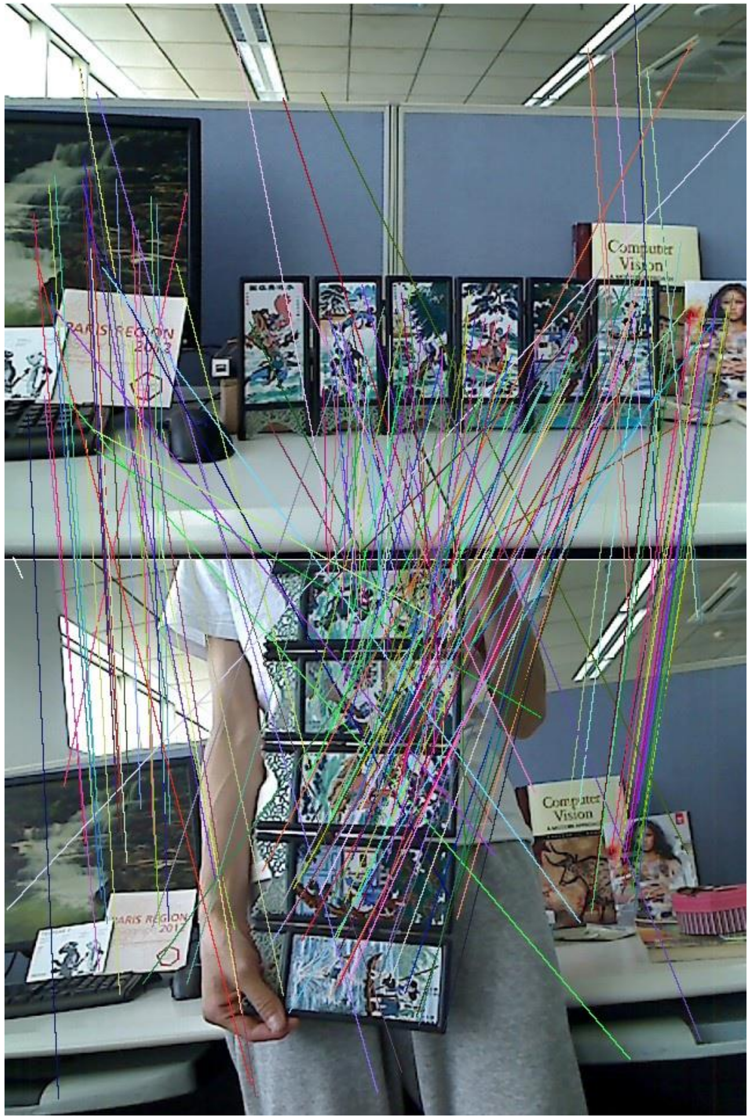
## Key Issues for SLAM in Dynamic Environments

- Gradually changing

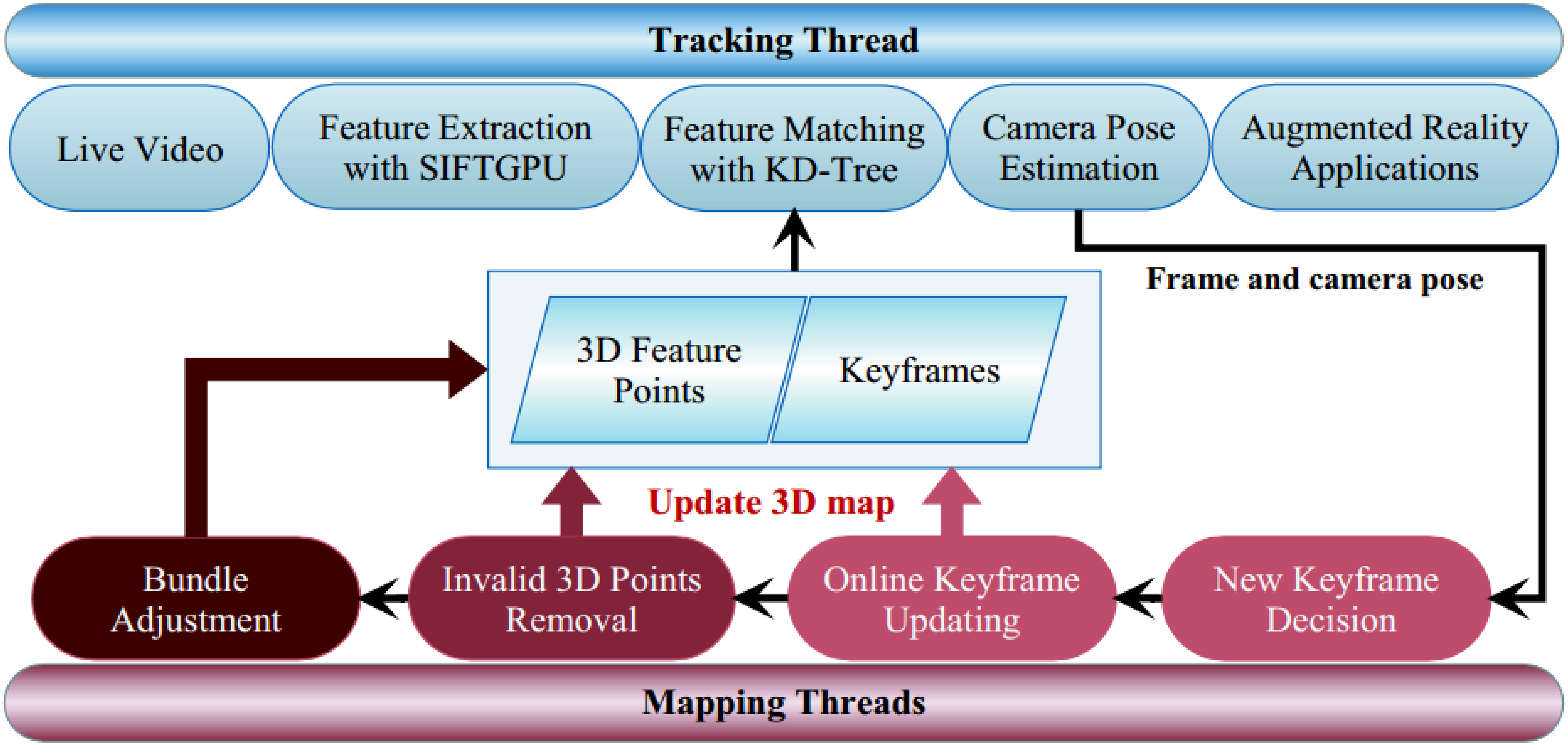
- Object Occlusion
  - Viewpoint Change
  - Dynamic Objects

## Key Issues for SLAM in Dynamic Environments

- Gradually changing
- Object Occlusion
    - Viewpoint Change
    - Dynamic Objects
- Very low inlier ratio

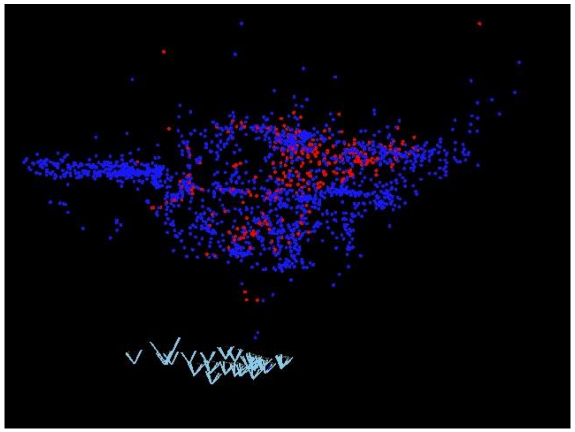


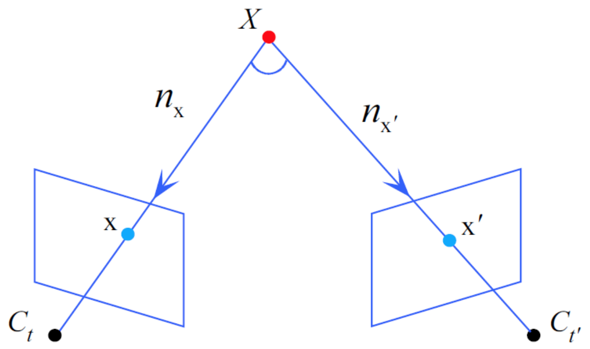
# RDSLAM Framework



# Online 3D Points and Keyframes Updating

- Keyframe representation
- 3D Change detection
  - Select 5 closest keyframes for online image.
  - For each valid feature point x in each selected keyframe,
    - Compute its projection x' in current frame $D_c(X) = \min_d \sum_{\mathbf{y} \in W(\mathbf{x})} |I_{\mathbf{y}} - I_{\mathbf{y}'+d}|$
    - If $n_{\mathbf{x}}^\top \hat{n}_{\mathbf{x}'} < \tau_n$ , compute the appearance difference
      - If $D_c(X) > \tau_c$ , then find a set of feature points y close to x'.
        - If $z_{X_{\mathbf{y}}} \geq z_X$ or their depths are very close, set V(X)=0.

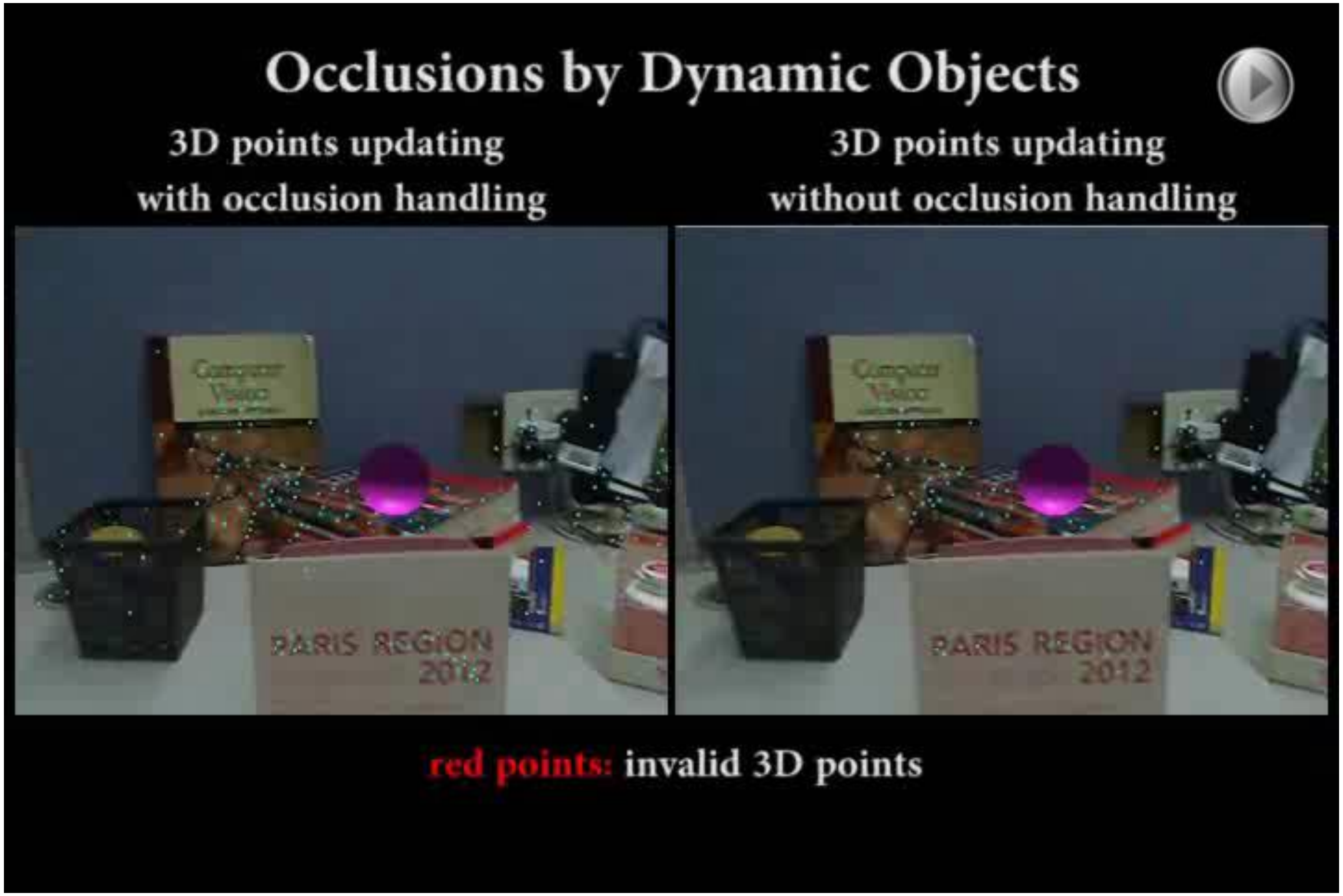




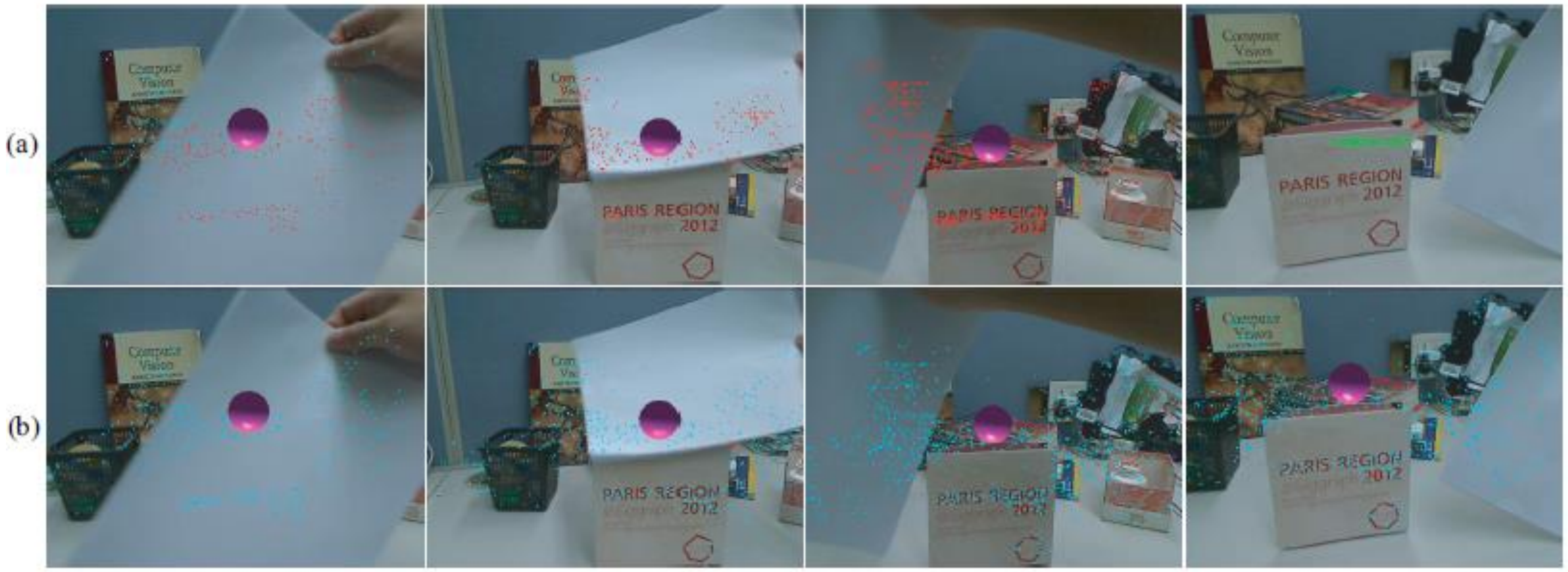
Since dynamic points cannot be triangulated, the occlusion caused by dynamic objects can be excluded here.

The occlusions caused by static objects are also excluded.

## Occlusion Handling


Occlusions by Dynamic Objects
3D points updating with occlusion handling
3D points updating without occlusion handling
red points: invalid 3D points

## Occlusion Handling



(a) The SLAM result without occlusion handling.
(b) The SLAM result with occlusion handling.

## Random Sample Consensus (RANSAC)

[Fischler and Bolles, 1981]
**Objective**: Robust fit of a model to a data set $S$ which contains outliers.

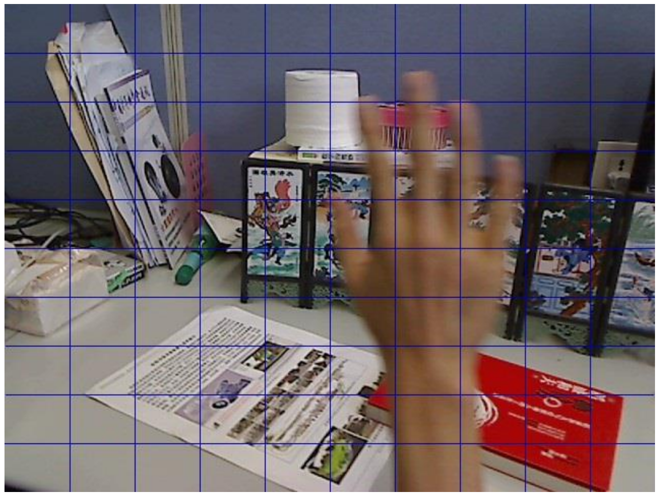
```
Step 1. Compute a set of potential matches
Step 2. While T(#inliers, #samples) < 95% do
        step 2.1 select minimal sample (6 matches)
        step 2.2 compute solutions for $P$
        step 2.3 determine inliers
Step 3. Refine $P$ based on all inliers
```

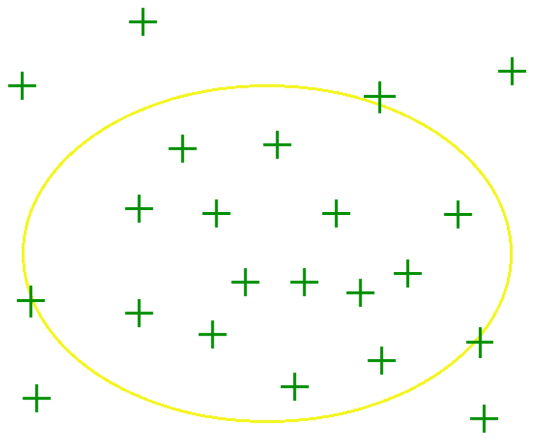
## Prior-based Adaptive RANSAC

- Sample generation
  - 10x10 bins
  - Prior probability $p_i = \varepsilon_i^* / \sum_j \varepsilon_j^*$
- Hypothesis evaluation

$$s = (\sum_i \varepsilon_i) \frac{\pi \sqrt{\det(C)}}{A}$$

  - Inliers number $N \approx \sum_i \varepsilon_i$
  - Inliers distribution, i.e., distribution ellipse $C$



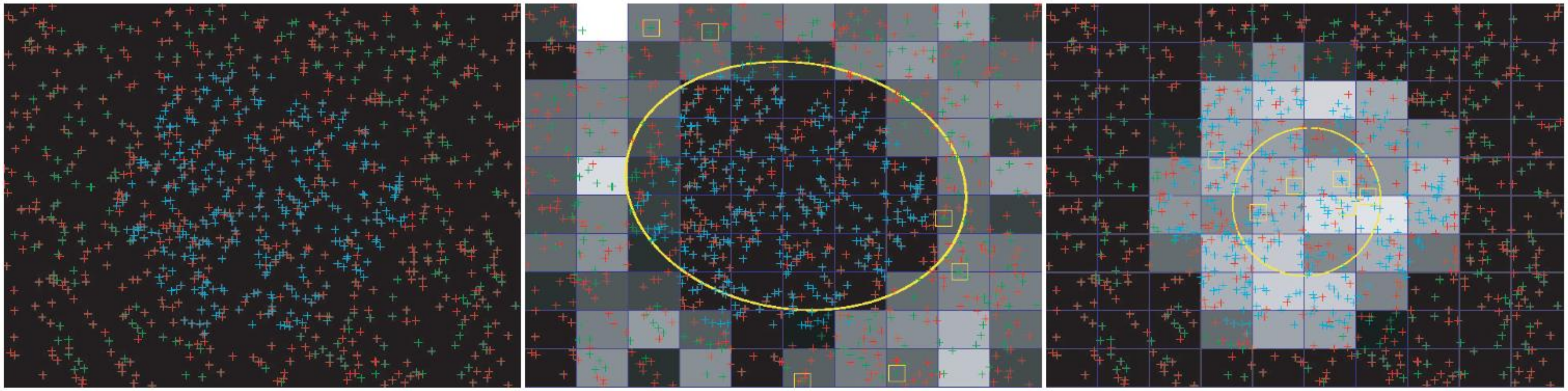


## Prior-based Adaptive RANSAC

- Hypothesis evaluation

$$s = (\sum_i \varepsilon_i)\frac{\pi\sqrt{\det(C)}}{A}$$

$\sum_i \varepsilon_i$=24.94 $\sum_i \varepsilon_i$=21.77



200 green points on the static background, 300 cyan points on the rigidly moving object, 500 red points are randomly moving.

## Prior-based Adaptive RANSAC

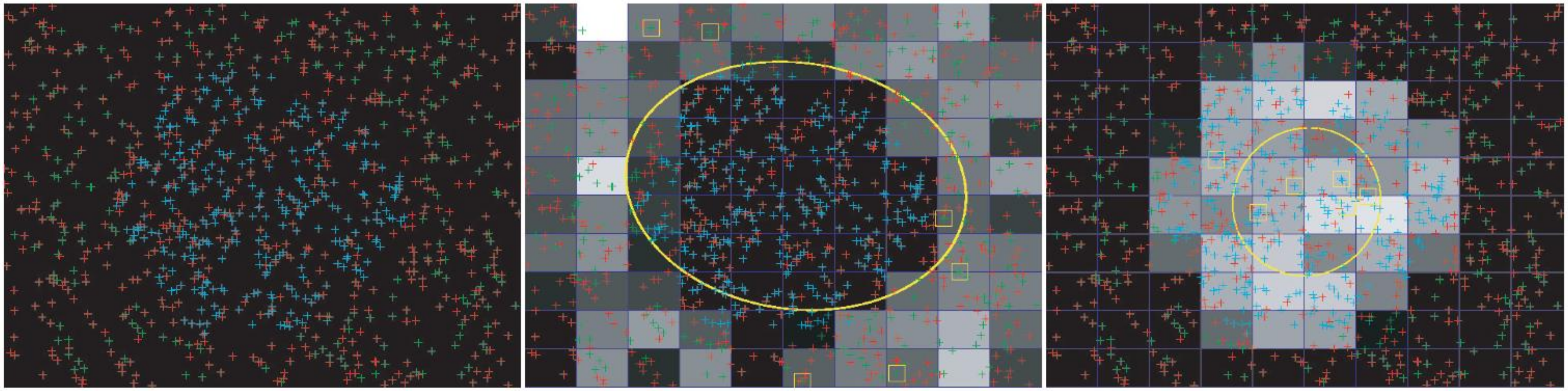
- Hypothesis evaluation

$$s = (\sum_i \varepsilon_i)\frac{\pi\sqrt{\det(C)}}{A}$$

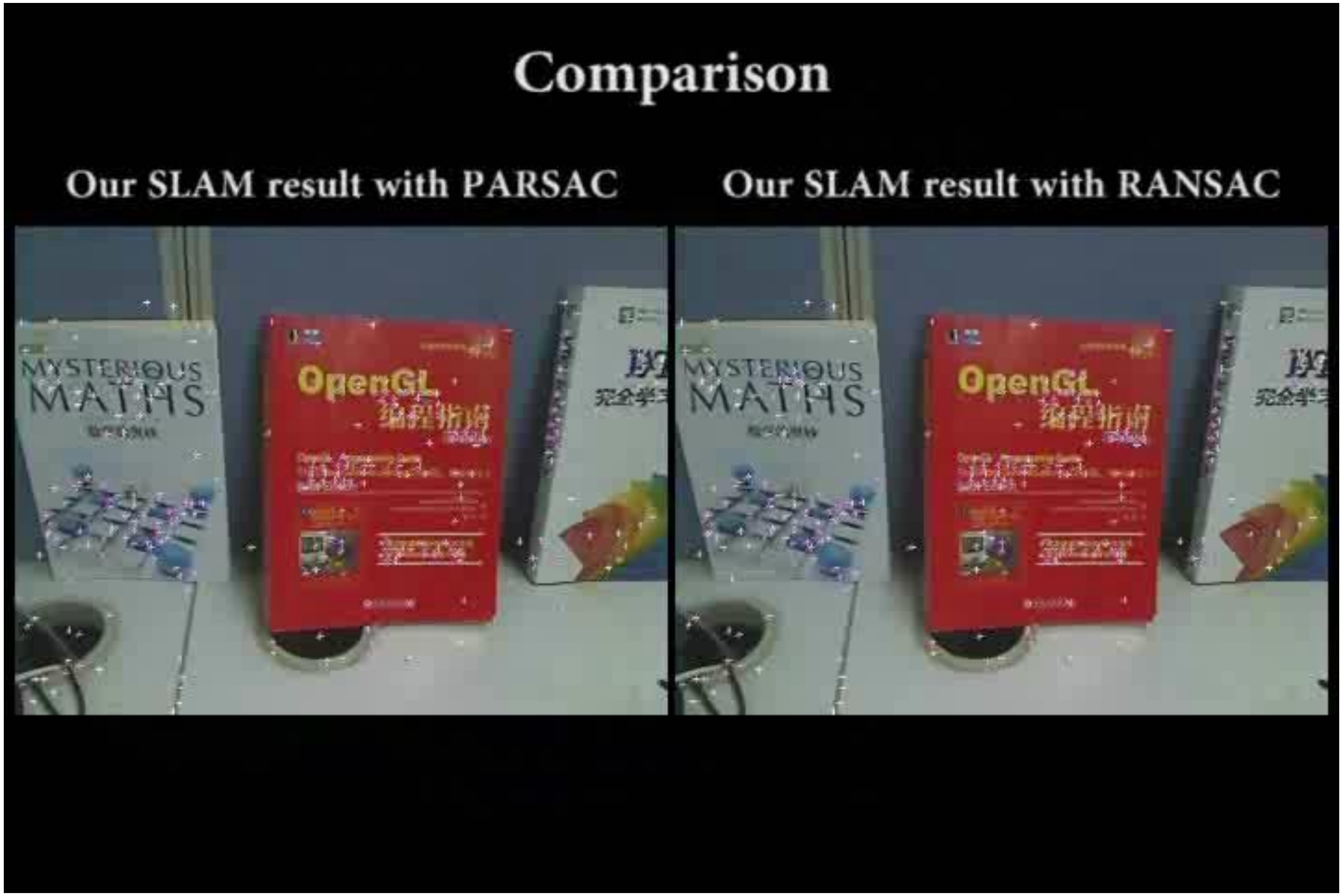
S1 = 8.31 > S2 = 1.98

$\sum_i \varepsilon_i$=24.94

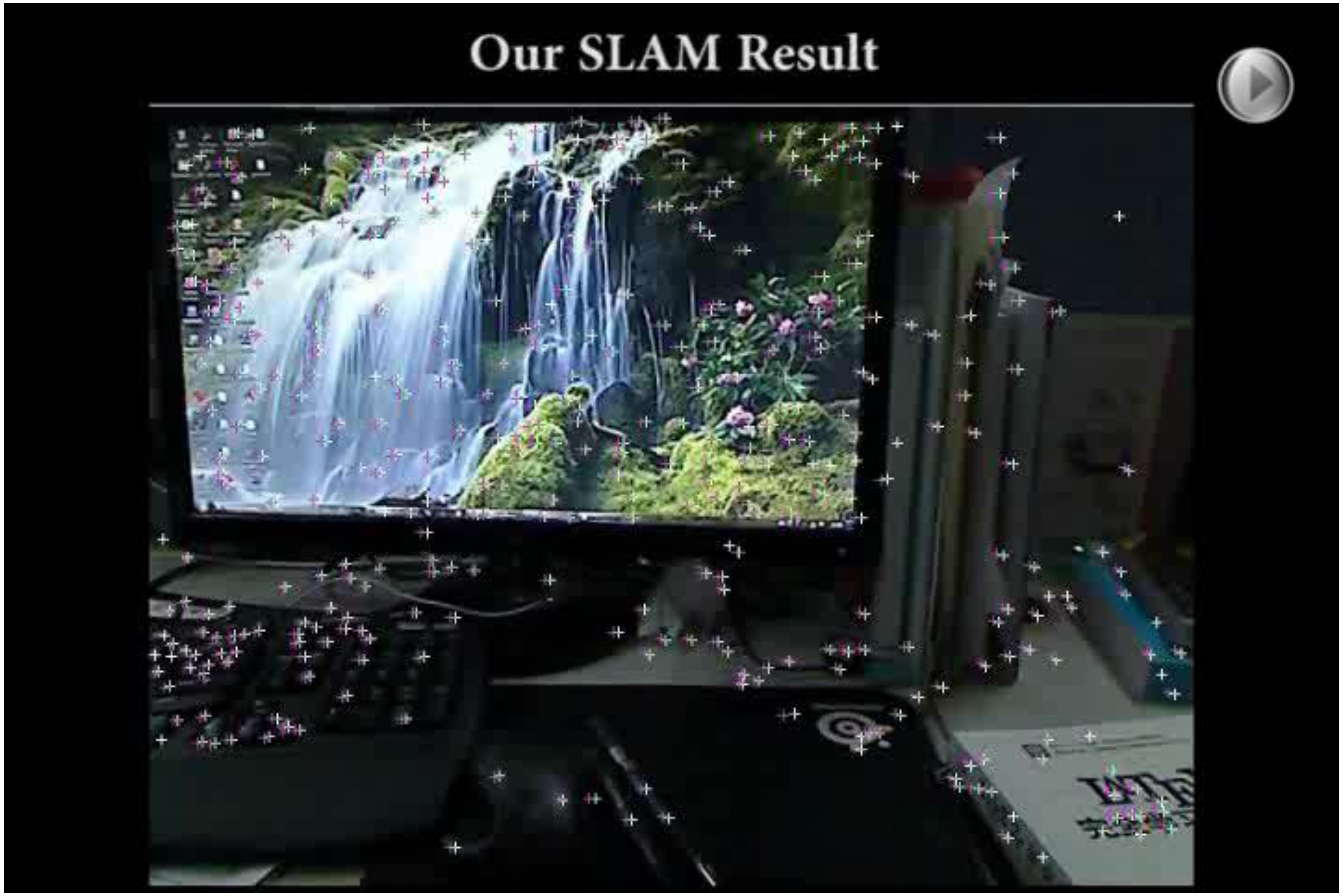
$\sum_i \varepsilon_i$=21.77


200 green points on the static background, 300 cyan points on the rigidly moving object, 500 red points are randomly moving.

## Results Comparison



## Results and Comparison



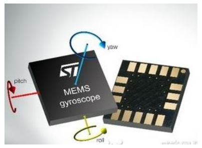
## Visual-Inertial SLAM

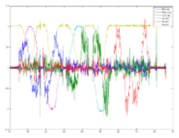
- Use IMU data to improve robustness
  - Filtering-based methods
    - MSCKF, …
  - Non-linear optimization based methods
    - OKVIS, VINS-Mono, …

- Can work without real IMU data?

# Optimization with Motion Priors

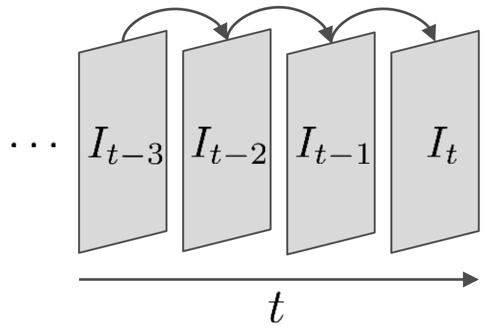
- **Sliding-window based pose optimization**
  - Motion constraints with IMU measurements





IMU measurements

$$\mathcal{X}^\star = \arg\min_{\mathcal{X}} \underbrace{\sum_{i,j} \|\mathbf{r}_{\mathcal{I}_{ij}}\|^2_{\Sigma_{ij}}}_{\text{IMU error}} + \underbrace{\sum_{i}\sum_{l} \|\mathbf{r}_{\mathcal{C}_{il}}\|^2_{\Sigma_C}}_{\text{reprojection error}}$$

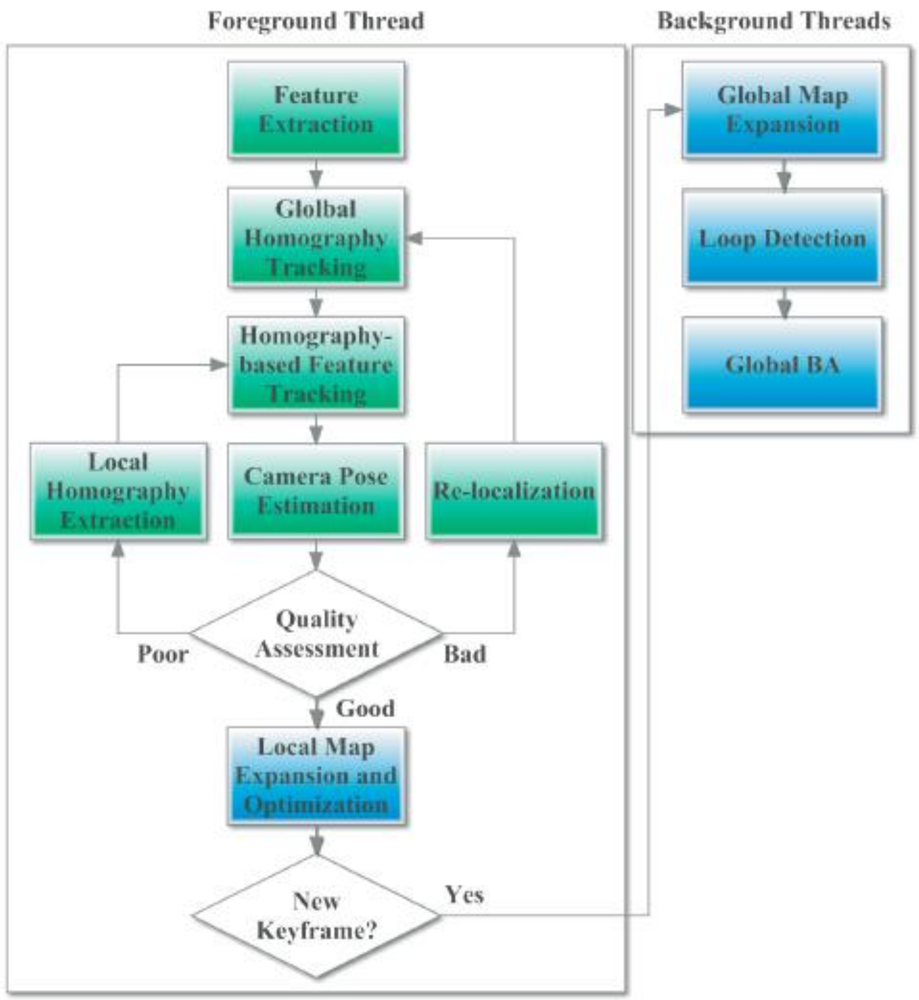


- **Motion constraints without IMU measurements**
  - Acceleration: is generally small and assumed to be zero
  - Rotational velocity: combine feature matching and global image alignment to estimate it

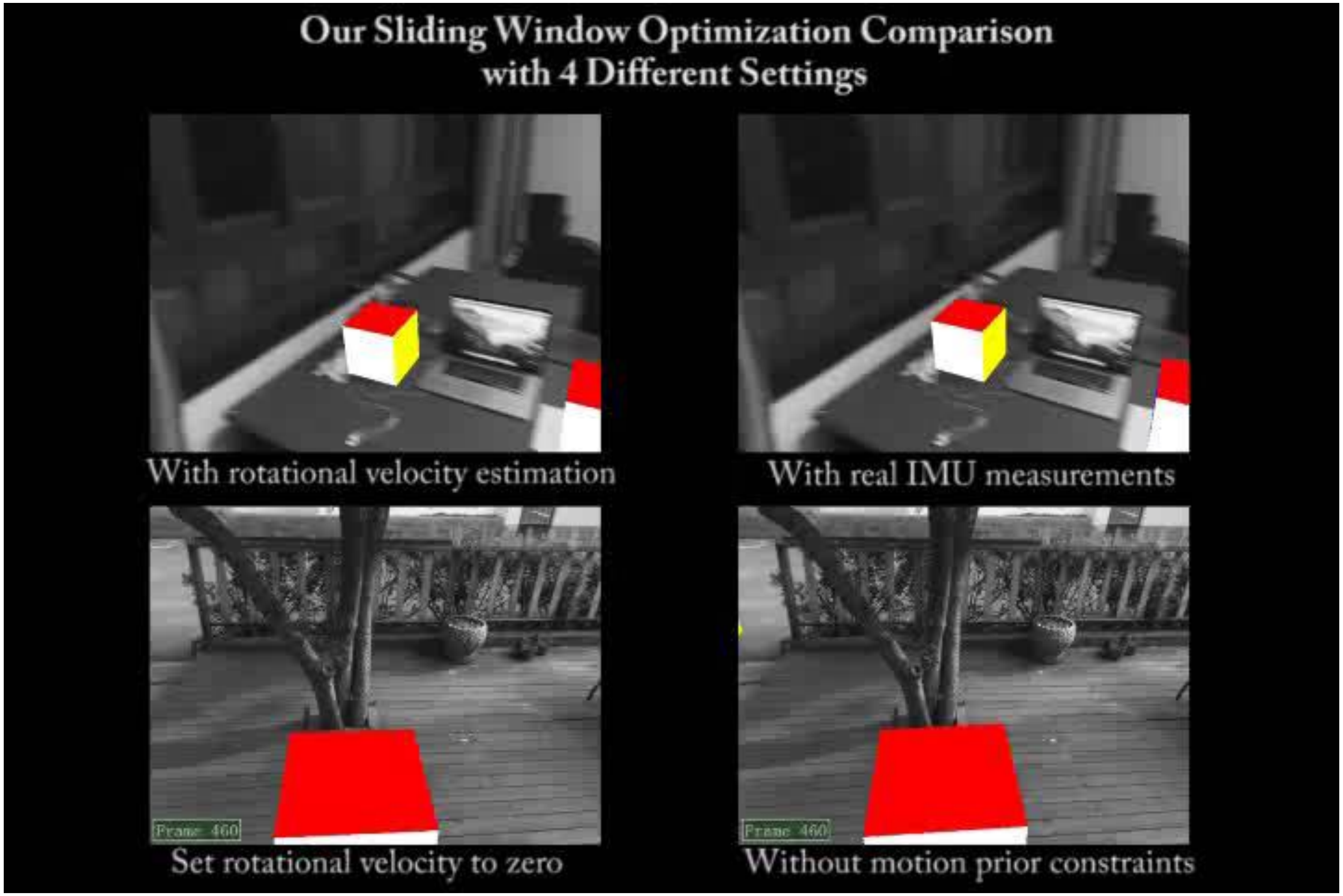
$$\hat{\omega}_i = \arg\min_{\omega} \Big( \sum_{x\in\mathbf{\Omega}} \|\tilde{I}_i(\mathbf{x}) - \tilde{I}_{i+1}(\pi(\mathbf{K}\mathbf{R}_\Delta(\boldsymbol{\omega}, t_{\Delta_i})\mathbf{K}^{-1}\mathbf{x}^h))\|_{\delta_\mathbf{I}}$$
$$+ \sum_{(\mathbf{x}_i,\mathbf{x}_{i+1})\in M_{i,i+1}} \frac{1}{\delta_\mathbf{X}} \|\pi(\mathbf{K}\mathbf{R}_\Delta(\boldsymbol{\omega}, t_{\Delta_i})\mathbf{K}^{-1}\mathbf{x}_i^h) - \mathbf{x}_{i+1}\|_2^2 \Big)$$

# RKSLAM Framework

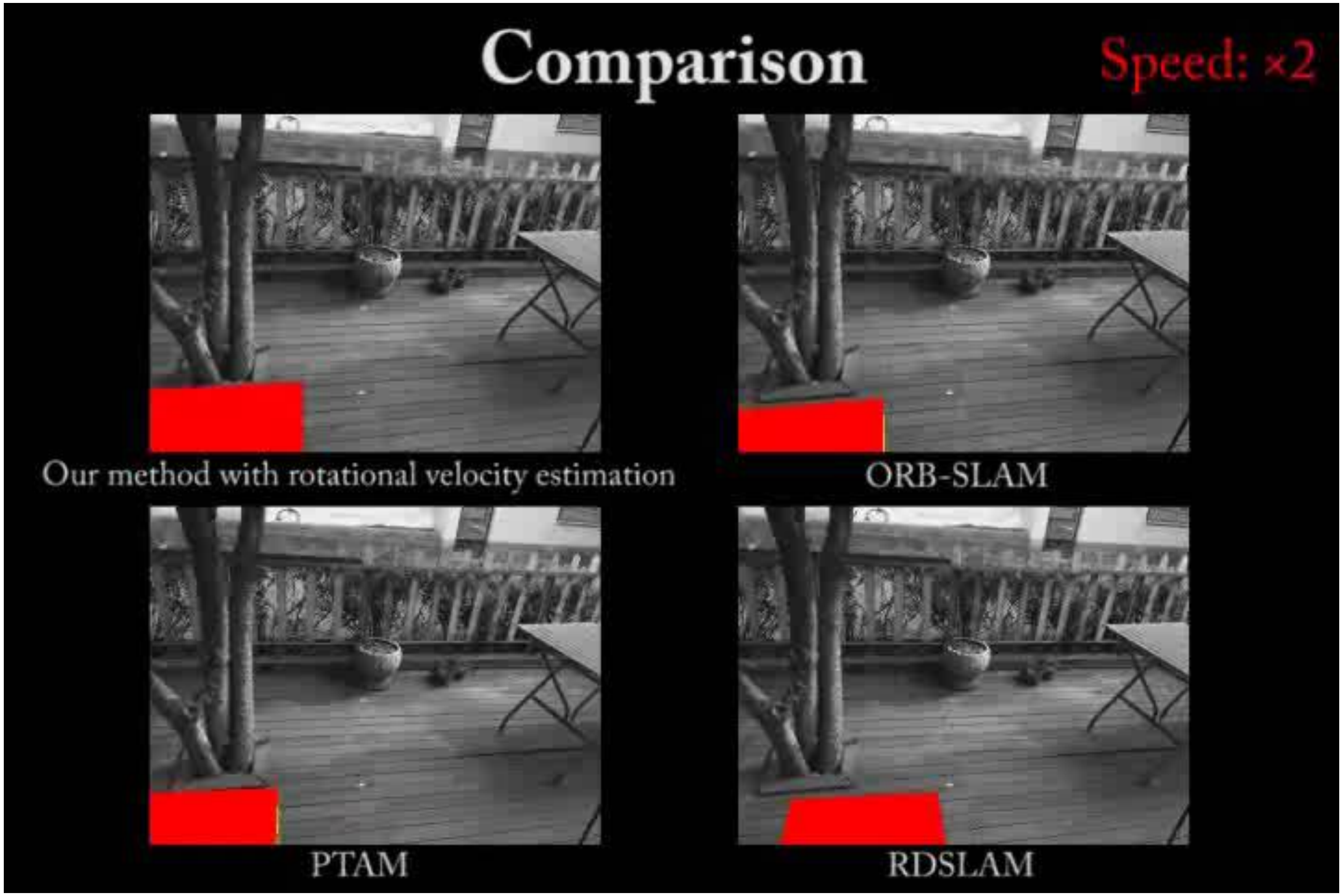
- Multi-Homography based Tracking
  - Global homography
  - Specific Homography
  - Local Homographies
- Sliding-window based pose optimization
  - Use global image alignment and feature matches to estimate rotational velocity.
  - Pose optimization with simulated IMU data.



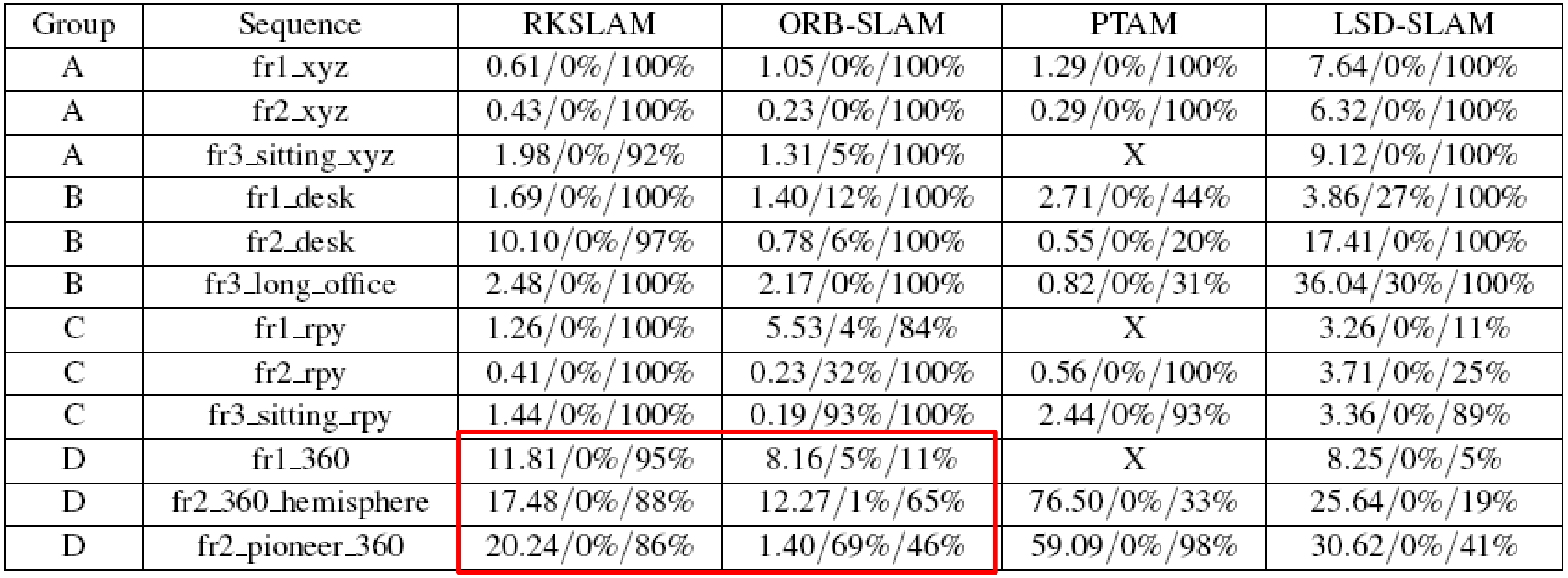
## Sliding-Window based Optimization Comparison


Our Sliding Window Optimization Comparison
with 4 Different Settings
With rotational velocity estimation
With real IMU measurements
Frame 460
Set rotational velocity to zero
Frame 460
Without motion prior constraints

## Results and Comparions



# Quantitative Evaluation with TUM RGB-D Dataset



| Group | Sequence | RKSLAM | ORB-SLAM | PTAM | LSD-SLAM |
|---|---|---|---|---|---|
| A | fr1_xyz | 0.61/0%/100% | 1.05/0%/100% | 1.29/0%/100% | 7.64/0%/100% |
| A | fr2_xyz | 0.43/0%/100% | 0.23/0%/100% | 0.29/0%/100% | 6.32/0%/100% |
| A | fr3_sitting_xyz | 1.98/0%/92% | 1.31/5%/100% | X | 9.12/0%/100% |
| B | fr1_desk | 1.69/0%/100% | 1.40/12%/100% | 2.71/0%/44% | 3.86/27%/100% |
| B | fr2_desk | 10.10/0%/97% | 0.78/6%/100% | 0.55/0%/20% | 17.41/0%/100% |
| B | fr3_long_office | 2.48/0%/100% | 2.17/0%/100% | 0.82/0%/31% | 36.04/30%/100% |
| C | fr1_rpy | 1.26/0%/100% | 5.53/4%/84% | X | 3.26/0%/11% |
| C | fr2_rpy | 0.41/0%/100% | 0.23/32%/100% | 0.56/0%/100% | 3.71/0%/25% |
| C | fr3_sitting_rpy | 1.44/0%/100% | 0.19/93%/100% | 2.44/0%/93% | 3.36/0%/89% |
| D | fr1_360 | 11.81/0%/95% | 8.16/5%/11% | X | 8.25/0%/5% |
| D | fr2_360_hemisphere | 17.48/0%/88% | 12.27/1%/65% | 76.50/0%/33% | 25.64/0%/19% |
| D | fr2_pioneer_360 | 20.24/0%/86% | 1.40/69%/46% | 59.09/0%/98% | 30.62/0%/41% |

From left to right: RMSE (cm) of keyframes, the starting ratio (i.e. dividing the initialization frame index by the total frame number), and the tracking success ratio after initialization.

Group A: simple translation

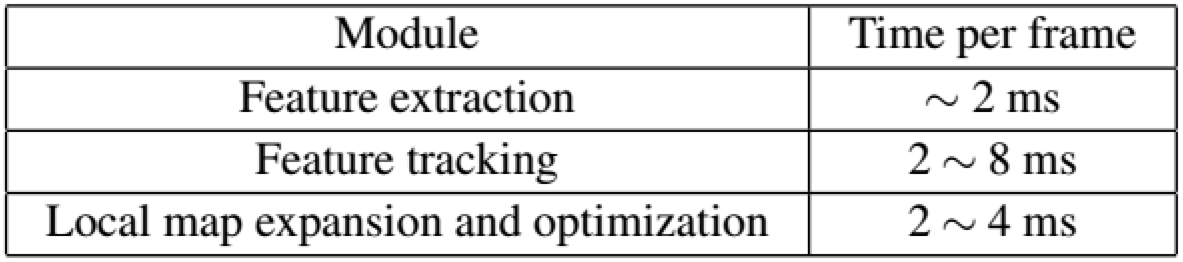
Group B: there are loops

Group C: slow and nearly pure rotation

Group D: fast motion with strong rotation

## Timing

- Computation Time on a desktop PC



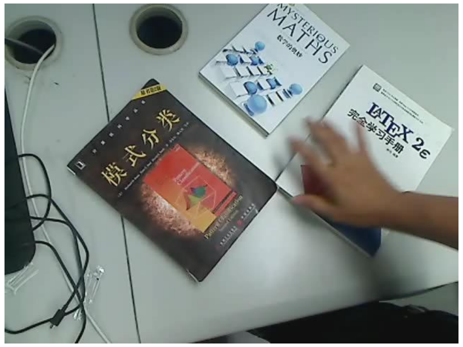
| Module | Time per frame |
|---|---|
| Feature extraction | $\sim 2$ ms |
| Feature tracking | $2 \sim 8$ ms |
| Local map expansion and optimization | $2 \sim 4$ ms |

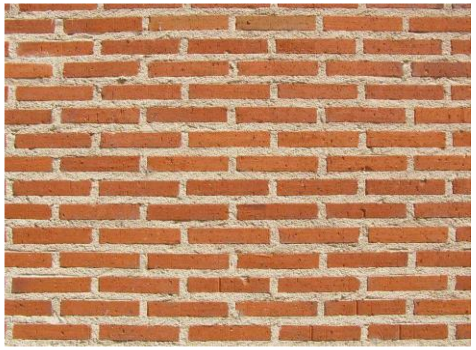
Table 1: Process time per frame with a single thread.

- For a mobile device
  - 20~50 fps on an iPhone 6.

## Major Challenges of SLAM for AR

- **Unexpected Situations in Applications**
  - A home user may not carefully move the AR device.
  - Real environment may have moving objects, large textureless/repeated regions, and strong occlusions.
- **Good User Experiences**
  - Accurate and consistent 3D registration.
  - Low frequency of camera lost.
  - Quick recovery from failure status.





# Visual-Inertial Dataset

- Typical VIO Dataset (e.g. EuRoC, TUM VI)
  - Synchronized sensors.
  - Global shutter cameras with high quality IMU.
- Mobile Phone Data
  - Sensor synchronization is not so reliable.
  - Rolling shutter camera with low-cost IMU.
- Not for evaluating real AR applications.

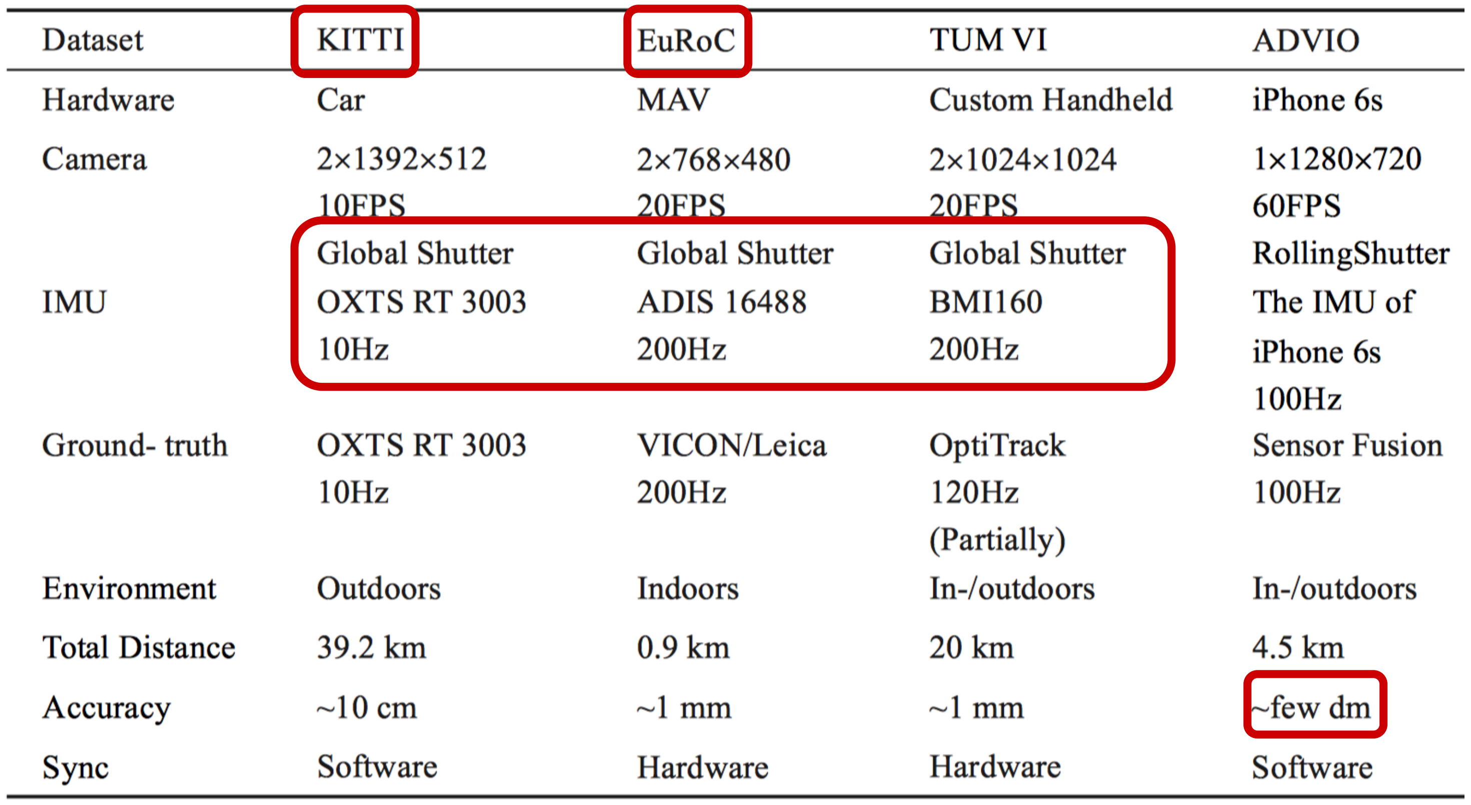
EuRoC

TUM VI

Real AR Application

# Visual-Inertial Dataset

Comparison of commonly used VISLAM datasets

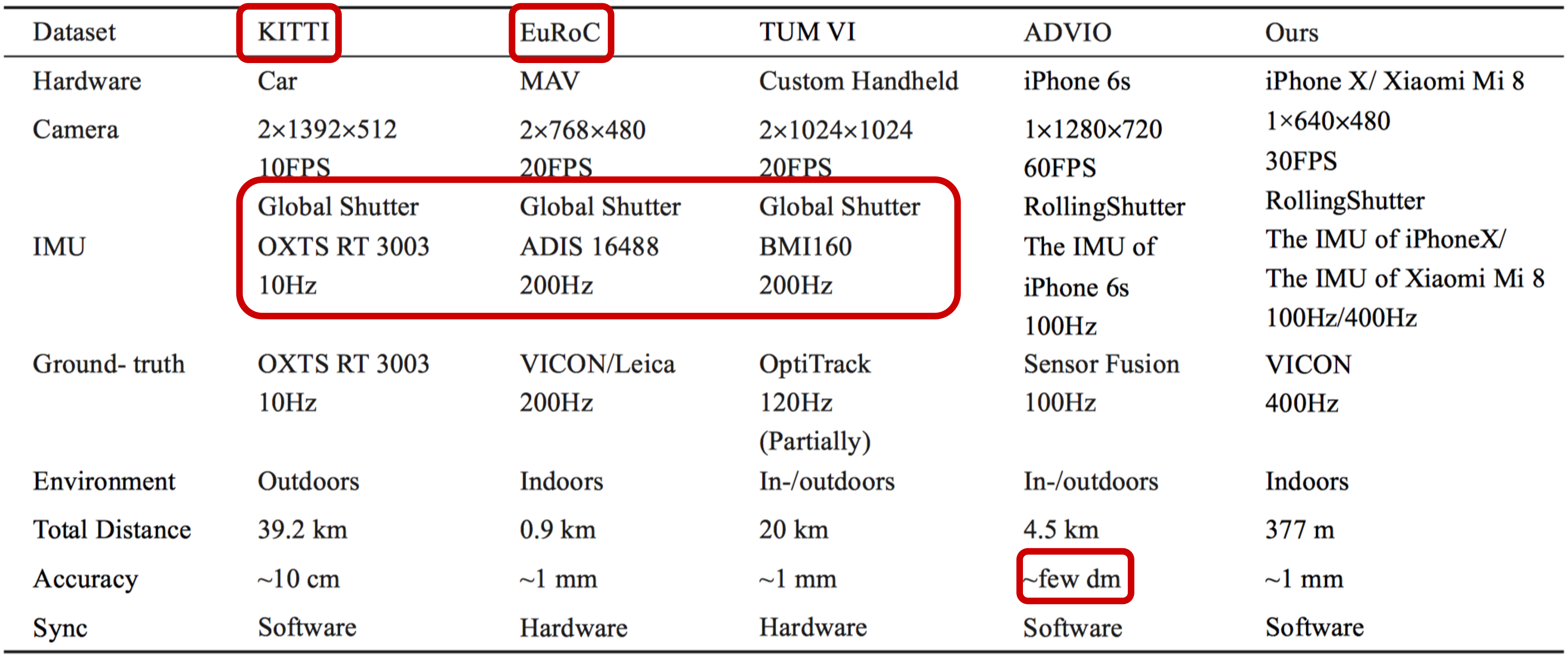


| Dataset | KITTI | EuRoC | TUM VI | ADVIO |
|---|---|---|---|---|
| Hardware | Car | MAV | Custom Handheld | iPhone 6s |
| Camera | 2×1392×512 10FPS Global Shutter | 2×768×480 20FPS Global Shutter | 2×1024×1024 20FPS Global Shutter | 1×1280×720 60FPS RollingShutter |
| IMU | OXTS RT 3003 10Hz | ADIS 16488 200Hz | BMI160 200Hz | The IMU of iPhone 6s 100Hz |
| Ground- truth | OXTS RT 3003 10Hz | VICON/Leica 200Hz | OptiTrack 120Hz (Partially) | Sensor Fusion 100Hz |
| Environment | Outdoors | Indoors | In-/outdoors | In-/outdoors |
| Total Distance | 39.2 km | 0.9 km | 20 km | 4.5 km |
| Accuracy | ~10 cm | ~1 mm | ~1 mm | ~few dm |
| Sync | Software | Hardware | Hardware | Software |

We need a more appropriate dataset for evaluating SLAM performance in AR applications, along with high accuracy ground-truth.

# Visual-Inertial Dataset

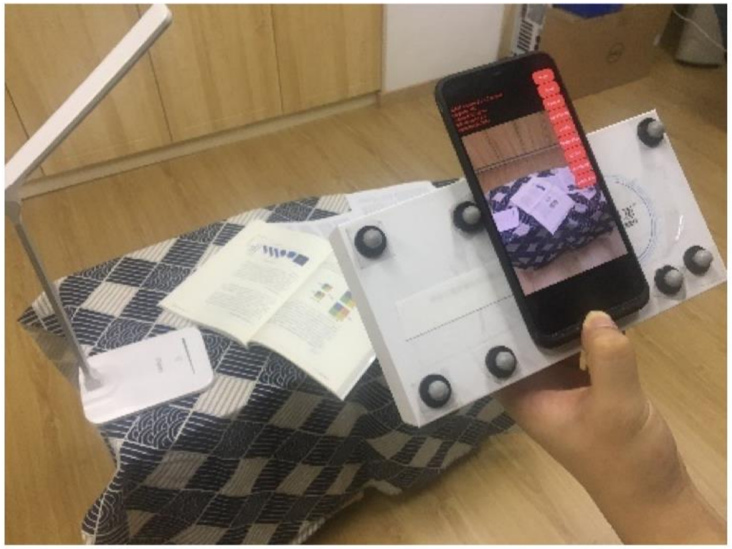
Comparison of commonly used VISLAM datasets

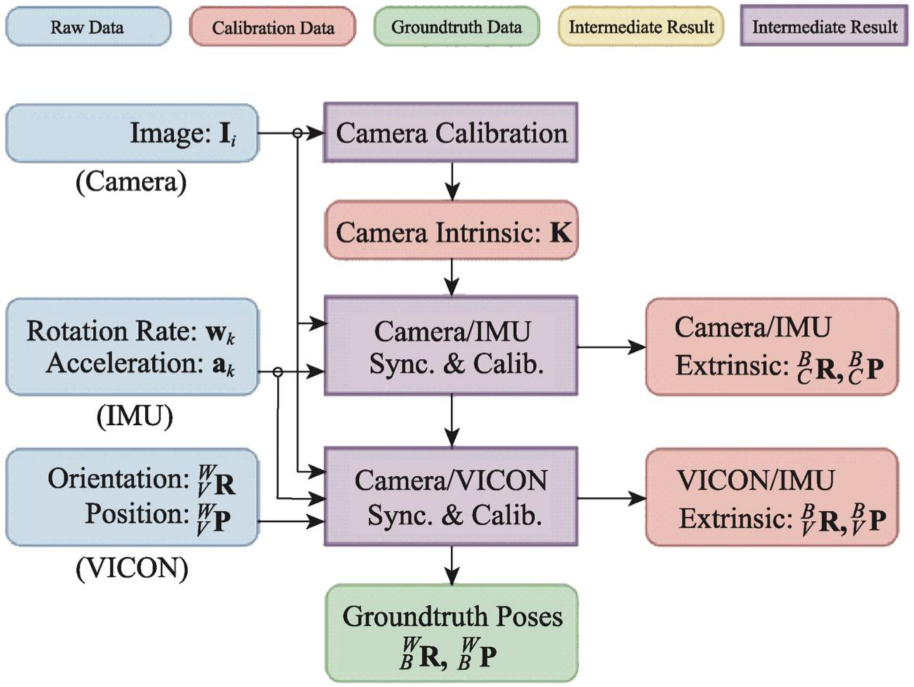


| Dataset | KITTI | EuRoC | TUM VI | ADVIO | Ours |
|---|---|---|---|---|---|
| Hardware | Car | MAV | Custom Handheld | iPhone 6s | iPhone X/ Xiaomi Mi 8 |
| Camera | 2×1392×512 10FPS Global Shutter | 2×768×480 20FPS Global Shutter | 2×1024×1024 20FPS Global Shutter | 1×1280×720 60FPS RollingShutter | 1×640×480 30FPS RollingShutter |
| IMU | OXTS RT 3003 10Hz | ADIS 16488 200Hz | BMI160 200Hz | The IMU of iPhone 6s 100Hz | The IMU of iPhoneX/ The IMU of Xiaomi Mi 8 100Hz/400Hz |
| Ground- truth | OXTS RT 3003 10Hz | VICON/Leica 200Hz | OptiTrack 120Hz (Partially) | Sensor Fusion 100Hz | VICON 400Hz |
| Environment | Outdoors | Indoors | In-/outdoors | In-/outdoors | Indoors |
| Total Distance | 39.2 km | 0.9 km | 20 km | 4.5 km | 377 m |
| Accuracy | ~10 cm | ~1 mm | ~1 mm | ~few dm | ~1 mm |
| Sync | Software | Hardware | Hardware | Software | Software |

# Hardware Setup & Data Process

- Two different mobile phones
    - iPhone X (Camera 640x480 30fps, IMU 100Hz)
    - Xiaomi Mi 8 (Camera 640x480 30fps, IMU 400Hz)
- Ground-truth obtained by VICON system at 400Hz

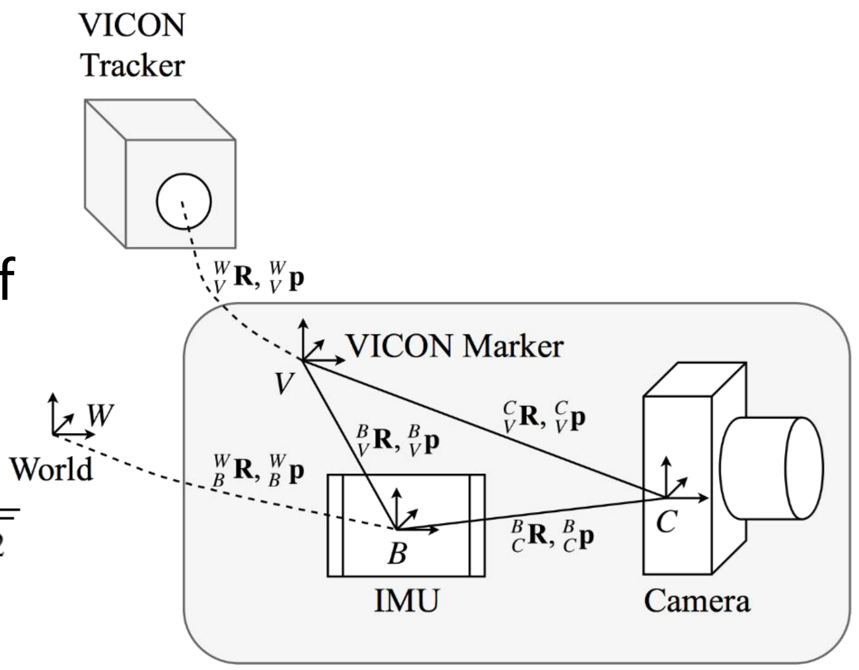

The phone is rigidly attached to a marker object for VICON localization


Raw Data
Calibration Data
Groundtruth Data
Intermediate Result
Intermediate Result
Image: $\mathbf{I}_i$
(Camera)
Camera Calibration
Camera Intrinsic: $\mathbf{K}$
Rotation Rate: $\mathbf{w}_k$
Acceleration: $\mathbf{a}_k$
(IMU)
Camera/IMU Sync. & Calib.
Camera/IMU Extrinsic: ${}^B_C\mathbf{R}, {}^B_C\mathbf{P}$
Orientation: ${}^W_V\mathbf{R}$
Position: ${}^W_V\mathbf{P}$
(VICON)
Camera/VICON Sync. & Calib.
VICON/IMU Extrinsic: ${}^B_V\mathbf{R}, {}^B_V\mathbf{P}$
Groundtruth Poses ${}^W_B\mathbf{R}, {}^W_B\mathbf{P}$

## Device Synchronization and Calibration

- Camera-IMU Synchronization and Calibration
  - MATLAB Toolbox & Kalibr.
- VICON-IMU Synchronization
  - Maximizes the cross-correlation between VICON and IMU angle of rotations.

$$\underset{{}^{B}_{V}t}{\arg\max} \frac{\sum \left\| \theta_V({}_{V}t) \right\| \left\| \theta_B({}_{V}t + {}^{B}_{V}t) \right\|}{\sqrt{\sum \left\| \theta_V({}_{V}t) \right\|^2} \sqrt{\sum \left\| \theta_B({}_{V}t + {}^{B}_{V}t) \right\|^2}}$$

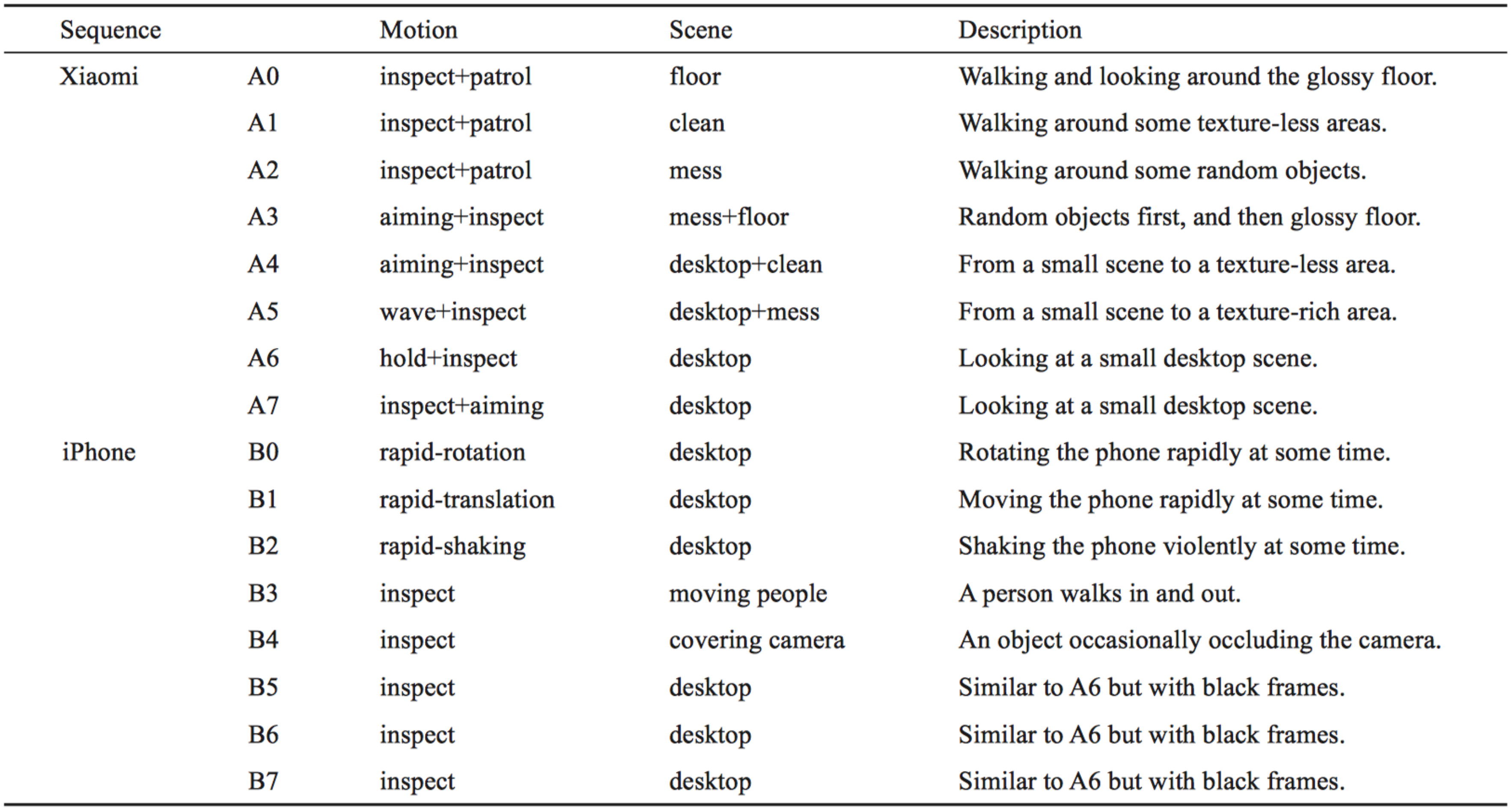

VICON Tracker
${}^{W}_{V}\mathbf{R}, {}^{W}_{V}\mathbf{p}$
VICON Marker
$V$
$W$
World
${}^{C}_{V}\mathbf{R}, {}^{C}_{V}\mathbf{p}$
${}^{B}_{V}\mathbf{R}, {}^{B}_{V}\mathbf{p}$
${}^{W}_{B}\mathbf{R}, {}^{W}_{B}\mathbf{p}$
$B$
IMU
${}^{B}_{C}\mathbf{R}, {}^{B}_{C}\mathbf{p}$
$C$
Camera

- VICON-Camera Calibration
  - Aligning the VICON measurements with the camera measurements by Apriltags.

$$\underset{\{{}^{W}_{V}\mathbf{R}_i\}, \{{}^{W}_{V}\mathbf{p}_i\}, {}^{C}_{V}\mathbf{R}, {}^{C}_{V}\mathbf{p}, \{\mathbf{X}_j\}}{\arg\min} \sum_i \left( \left\| \log\left({}^{W}_{V}\tilde{\mathbf{R}}_i \cdot {}^{W}_{V}\mathbf{R}_i\right) \right\|^2_{\Sigma_\theta} + \left\| {}^{W}_{V}\mathbf{p}_i - {}^{W}_{V}\tilde{\mathbf{p}}_i \right\|^2_{\Sigma_p} \right.$$
$$\left. + \sum_j \rho \left( \left\| \pi\left( {}^{C}_{V}\mathbf{R}\, {}^{W}_{V}\mathbf{R}_i \left( \mathbf{X}_j - {}^{W}_{V}\mathbf{p}_i \right) + {}^{C}_{V}\mathbf{p} \right) - \mathbf{x}_{ij} \right\|^2_{\Sigma_x} \right) \right)$$

## Dataset Motion and Scene Type

- 5 motion types : hold, wave, aiming, inspect, petrol
- 5 scene types : mess, clean, desktop, floor
- 3 segments : static, initialization, main
- B0~B7 are captured for evaluating dedicated criteria



| Sequence | | Motion | Scene | Description |
|---|---|---|---|---|
| Xiaomi | A0 | inspect+patrol | floor | Walking and looking around the glossy floor. |
| | A1 | inspect+patrol | clean | Walking around some texture-less areas. |
| | A2 | inspect+patrol | mess | Walking around some random objects. |
| | A3 | aiming+inspect | mess+floor | Random objects first, and then glossy floor. |
| | A4 | aiming+inspect | desktop+clean | From a small scene to a texture-less area. |
| | A5 | wave+inspect | desktop+mess | From a small scene to a texture-rich area. |
| | A6 | hold+inspect | desktop | Looking at a small desktop scene. |
| | A7 | inspect+aiming | desktop | Looking at a small desktop scene. |
| iPhone | B0 | rapid-rotation | desktop | Rotating the phone rapidly at some time. |
| | B1 | rapid-translation | desktop | Moving the phone rapidly at some time. |
| | B2 | rapid-shaking | desktop | Shaking the phone violently at some time. |
| | B3 | inspect | moving people | A person walks in and out. |
| | B4 | inspect | covering camera | An object occasionally occluding the camera. |
| | B5 | inspect | desktop | Similar to A6 but with black frames. |
| | B6 | inspect | desktop | Similar to A6 but with black frames. |
| | B7 | inspect | desktop | Similar to A6 but with black frames. |

## Dataset Preview

## Tracking Accuracy

- **4 commonly used criteria:**
  - Absolute Positional Error (APE)  Absolute Rotational Error (ARE)

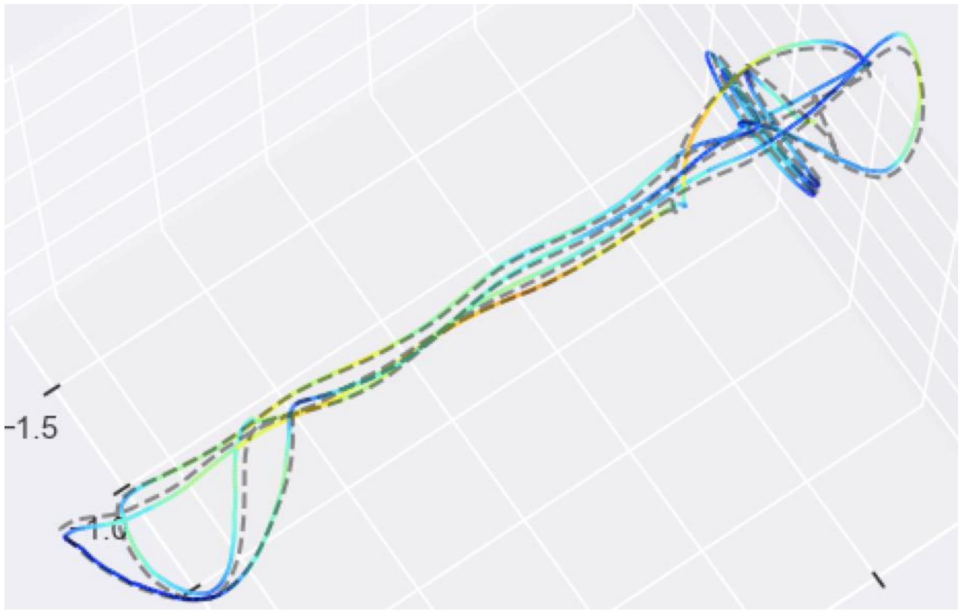
$$\epsilon_{\mathrm{APE}} = \sqrt{\frac{1}{m}\sum_{i=1}^{m} \| \mathbf{p}_{\mathrm{SLAM}}[i] - \mathbf{p}_{\mathrm{GT}}[i] \|^2}$$

$$\epsilon_{\mathrm{ARE}} = \sqrt{\frac{1}{m}\sum_{i=1}^{m} \| \log(\mathbf{R}^{-1}_{\mathrm{SLAM}}[i] \cdot \mathbf{R}_{\mathrm{GT}}[i]) \|^2}$$

  - Relative Positional Error (RPE)  Relative Rotational Error (RRE)

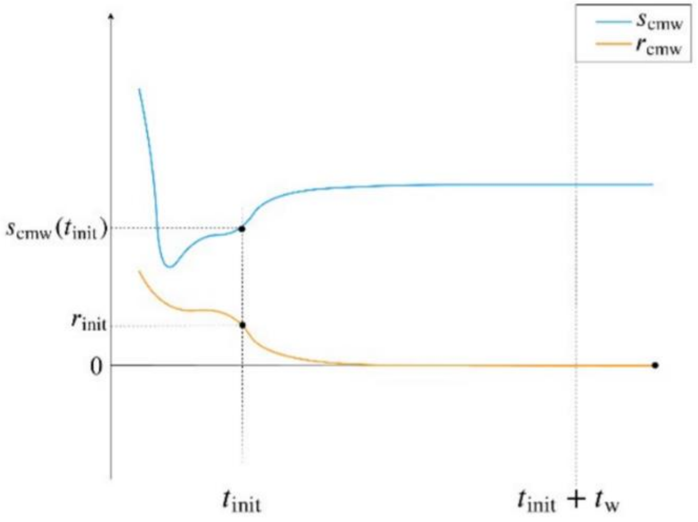
- **Completeness**
  - The ratio between the number of good poses and the total number of all poses.
  - The poses before the first initialization are not included.

Good Pose: APE $\leq 0.1$ m



## Initialization Quality

- The time $t_{\text{init}}$ for scale to converge.
  - Accurate scale is quite important in some AR applications.
  - At the beginning, scale usually fluctuates.
  - We generally insert AR objects after scale converges.
- The quality $\epsilon_{\text{scale}}$ of converged scale.
  - Key to some applications like AR ruler.
- For VSLAM, true scale is not available.
  - Estimate the global scale by aligning the results with ground-truth.



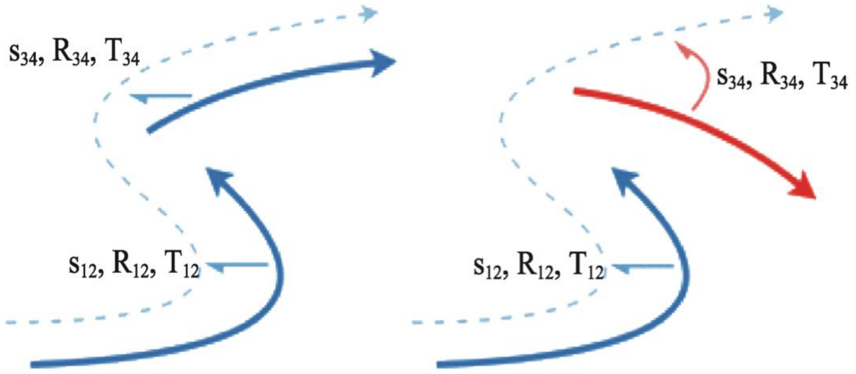
$$\epsilon_{\text{scale}} = \frac{1}{2}\left(\left|\frac{s_{\text{cmw}}(t_{\text{init}})}{s_g} - 1\right| + \left|\frac{s_g}{s_{\text{cmw}}(t_{\text{init}})} - 1\right|\right) \times 100\%$$

$$\epsilon_{\text{init}} = t_{\text{init}}(\epsilon_{\text{scale}} + \beta)^{\alpha}$$

## Tracking Robustness

- Relocalization Error
  - The tracking result should be consistent after recovering from lost status.

$$\epsilon_{RL} = \sum_{i=1}^{n-1} \| \log_{\mathrm{Sim}(3)}(\xi_i^{-1}\xi_{i+1}) \|$$



- Lost time: the smaller, the better.
- Smaller tracking error is better.

$$\epsilon_R = (\alpha_{\mathrm{lost}} + \eta_{\mathrm{lost}})(\epsilon_{RL} + \eta_{\mathrm{APE}}\epsilon_{\mathrm{APE}})$$

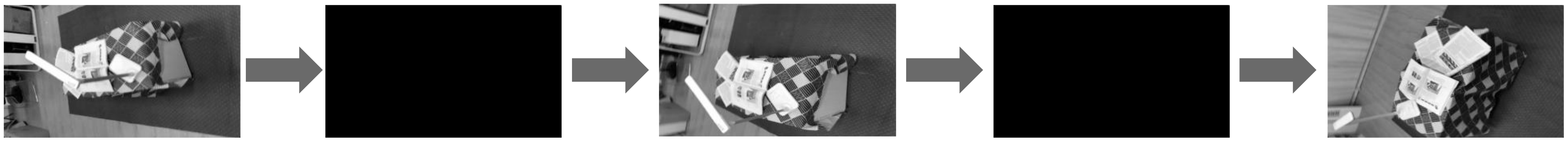
Ratio of Lost Time

APE

## Relocalization Time

- Force to enter lost state
  - Manually add black frames.
- Relocalization time measurement
  - VISLAM tends to continue IMU propagation even without sufficient feature matches.
  - Detect relocalization by the jump in trajectory.

$$t_{\mathrm{SLAM}[i]} \equiv \min\left\{t_k > t_{\mathrm{K}[i]} \mid \left\|\mathbf{p}_{\mathrm{SLAM}}[k+1]-\mathbf{p}_{\mathrm{SLAM}}[k]\right\| > \delta\right\}$$



## Representative SLAM Systems

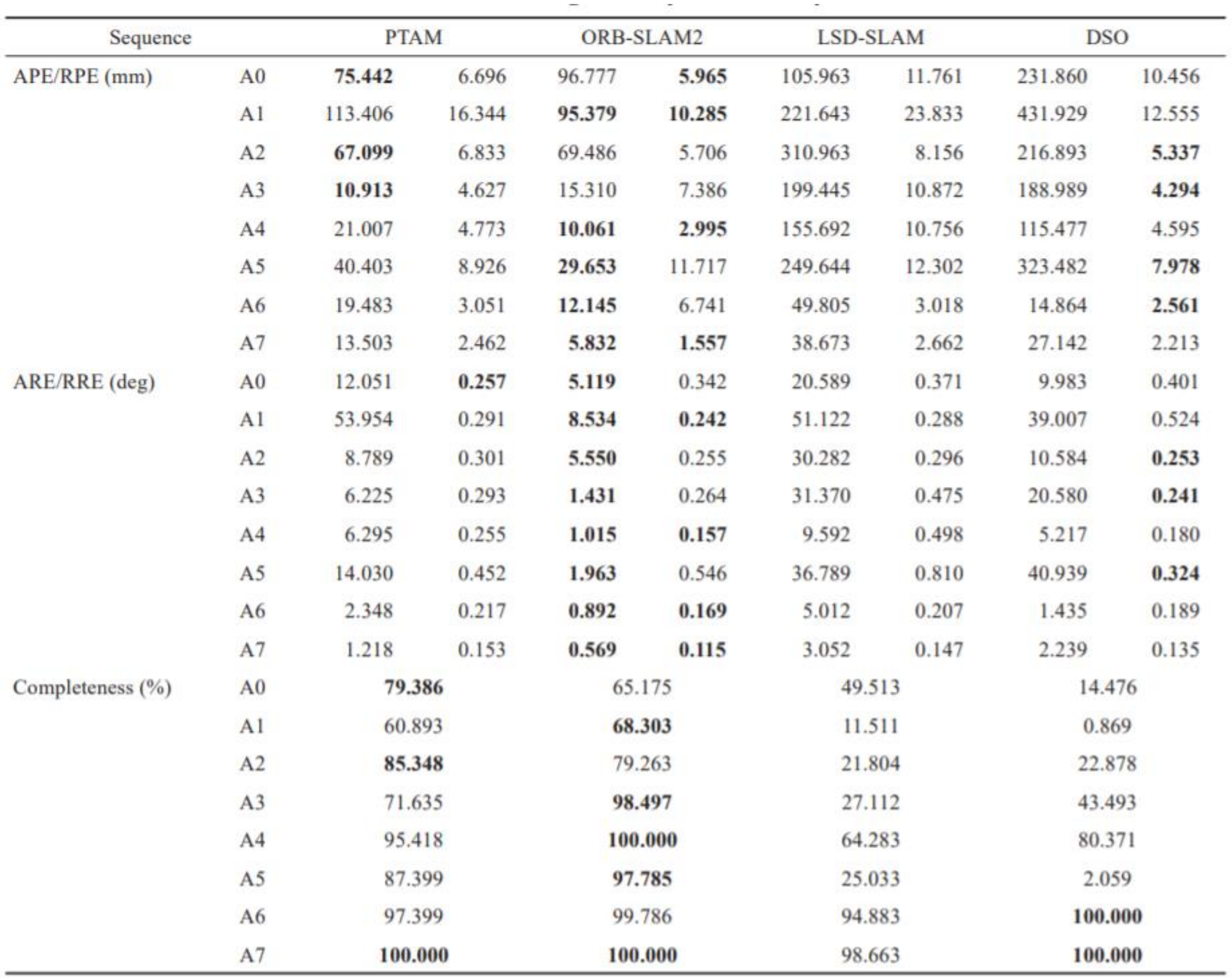
- Filtering-based SLAM
  - MonoSLAM : solve camera pose via extended Kalman filter.
  - MSCKF : keep a sliding window of $M$ frames.
  - MSCKF 2.0 : use FEJ to avoid leaking errors.
- Optimization-based SLAM
  - PTAM : use keyframe-based optimization, local tracking and global mapping in two parallel threads.
  - ORB-SLAM2 : use ORB features to improve the system robustness.
  - OKVIS : use sliding-window optimization with both reprojection errors and IMU motion errors.
  - VINS-Mono : use local sliding-window optimization and global pose graph optimization.
- SLAM with Direct Tracking
  - LSD-SLAM, DSO : directly use intensity as measurements and minimize photometric error.

## 8 Selected VSLAM/VSLAM Systems

- **VSLAM**
  - PTAM : http://wiki.ros.org/ethzasl_ptam
  - ORB-SLAM2 : https://github.com/raulmur/ORB_SLAM2
  - LSD-SLAM : https://github.com/tum-vision/lsd_slam
  - DSO : https://github.com/JakobEngel/dso
- **VISLAM**
  - MSCKF : https://github.com/daniilidis-group/msckf_mono
  - OKVIS : https://github.com/ethz-asl/okvis
  - VINS-Mono : https://github.com/HKUST-Aerial-Robotics/VINS-Mono
  - SenseSLAM v1.0: http://www.zjucvg.net/senseslam

# Experimental Results

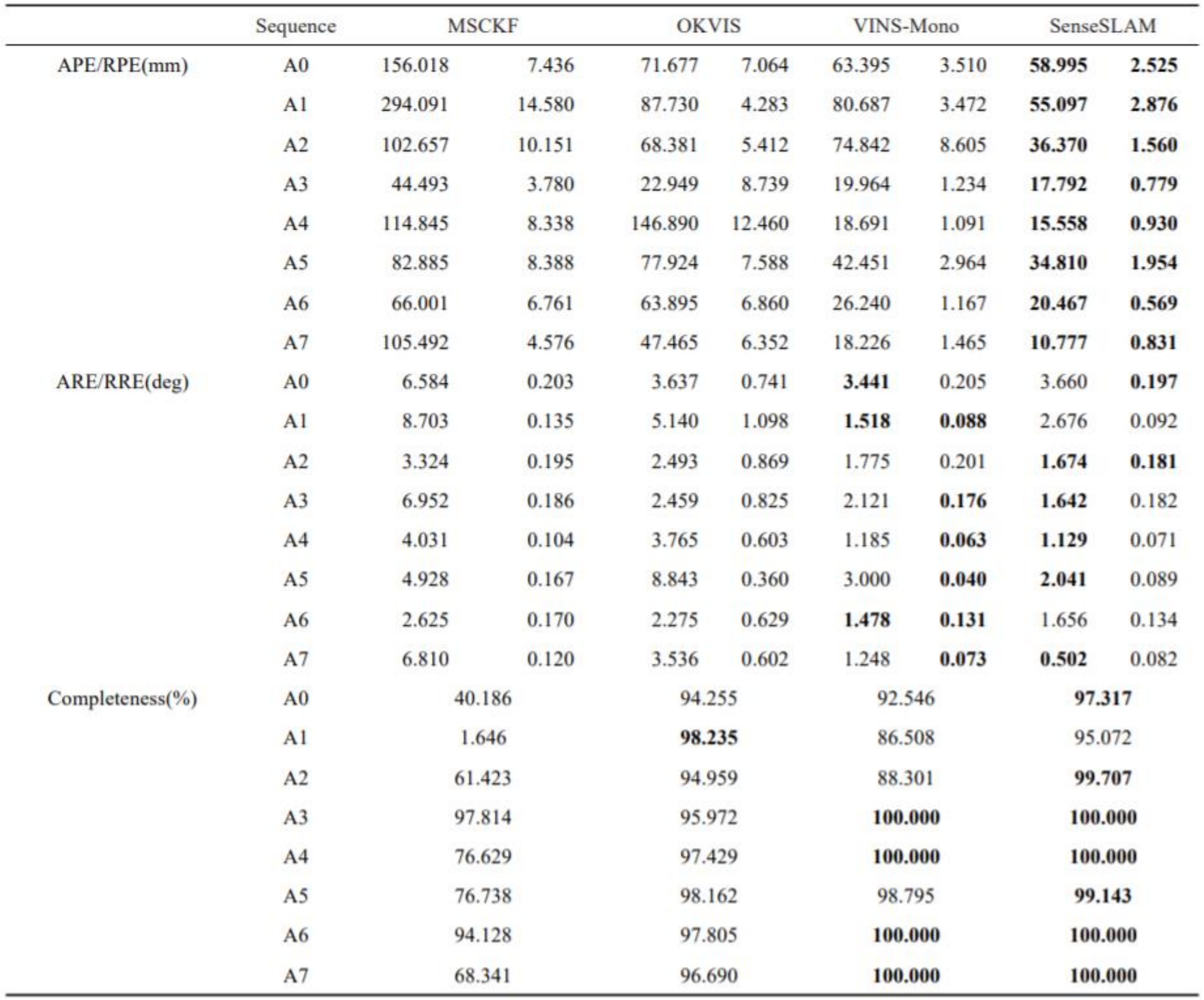
- VSLAM Tracking Accuracy



| Sequence | | PTAM | | ORB-SLAM2 | | LSD-SLAM | | DSO | |
|---|---|---|---|---|---|---|---|---|---|
| APE/RPE (mm) | A0 | **75.442** | 6.696 | 96.777 | **5.965** | 105.963 | 11.761 | 231.860 | 10.456 |
| | A1 | 113.406 | 16.344 | **95.379** | **10.285** | 221.643 | 23.833 | 431.929 | 12.555 |
| | A2 | **67.099** | 6.833 | 69.486 | 5.706 | 310.963 | 8.156 | 216.893 | **5.337** |
| | A3 | **10.913** | 4.627 | 15.310 | 7.386 | 199.445 | 10.872 | 188.989 | **4.294** |
| | A4 | 21.007 | 4.773 | **10.061** | **2.995** | 155.692 | 10.756 | 115.477 | 4.595 |
| | A5 | 40.403 | 8.926 | **29.653** | 11.717 | 249.644 | 12.302 | 323.482 | **7.978** |
| | A6 | 19.483 | 3.051 | **12.145** | 6.741 | 49.805 | 3.018 | 14.864 | **2.561** |
| | A7 | 13.503 | 2.462 | **5.832** | **1.557** | 38.673 | 2.662 | 27.142 | 2.213 |
| ARE/RRE (deg) | A0 | 12.051 | **0.257** | **5.119** | 0.342 | 20.589 | 0.371 | 9.983 | 0.401 |
| | A1 | 53.954 | 0.291 | **8.534** | **0.242** | 51.122 | 0.288 | 39.007 | 0.524 |
| | A2 | 8.789 | 0.301 | **5.550** | 0.255 | 30.282 | 0.296 | 10.584 | **0.253** |
| | A3 | 6.225 | 0.293 | **1.431** | 0.264 | 31.370 | 0.475 | 20.580 | **0.241** |
| | A4 | 6.295 | 0.255 | **1.015** | **0.157** | 9.592 | 0.498 | 5.217 | 0.180 |
| | A5 | 14.030 | 0.452 | **1.963** | 0.546 | 36.789 | 0.810 | 40.939 | **0.324** |
| | A6 | 2.348 | 0.217 | **0.892** | **0.169** | 5.012 | 0.207 | 1.435 | 0.189 |
| | A7 | 1.218 | 0.153 | **0.569** | **0.115** | 3.052 | 0.147 | 2.239 | 0.135 |
| Completeness (%) | A0 | **79.386** | | 65.175 | | 49.513 | | 14.476 | |
| | A1 | 60.893 | | **68.303** | | 11.511 | | 0.869 | |
| | A2 | **85.348** | | 79.263 | | 21.804 | | 22.878 | |
| | A3 | 71.635 | | **98.497** | | 27.112 | | 43.493 | |
| | A4 | 95.418 | | **100.000** | | 64.283 | | 80.371 | |
| | A5 | 87.399 | | **97.785** | | 25.033 | | 2.059 | |
| | A6 | 97.399 | | 99.786 | | 94.883 | | **100.000** | |
| | A7 | **100.000** | | **100.000** | | 98.663 | | **100.000** | |

# Experimental Results

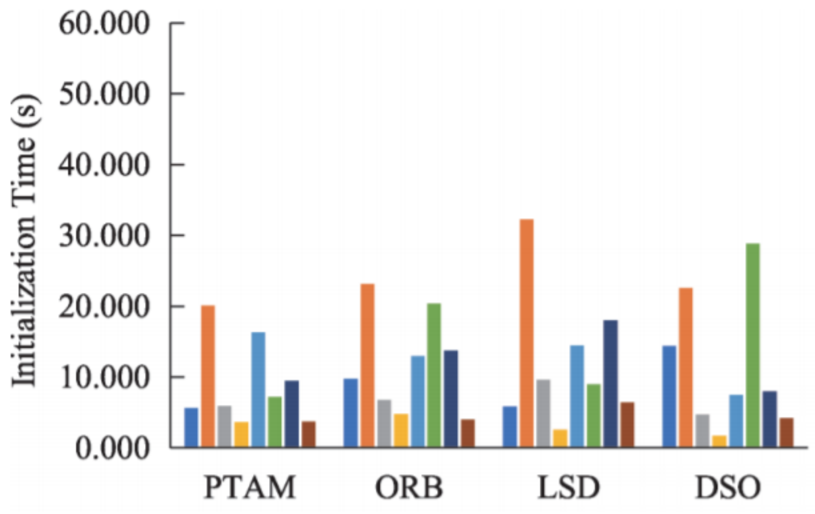
- VSLAM Tracking Accuracy

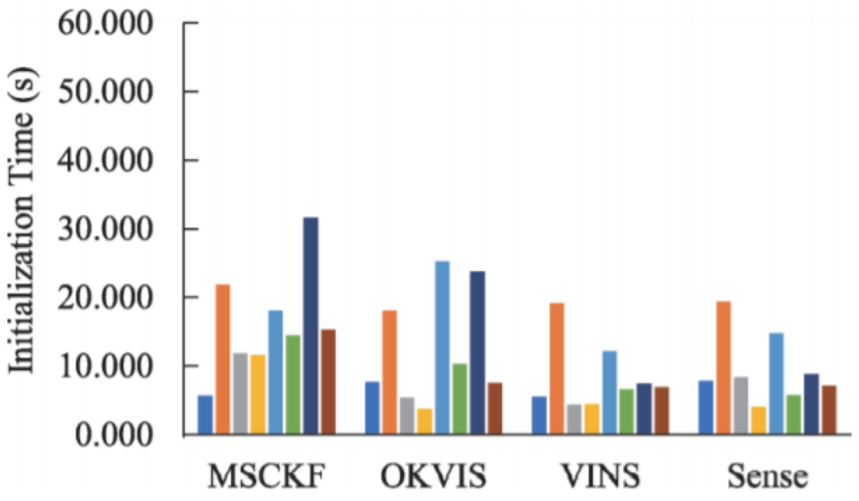


| | Sequence | MSCKF | | OKVIS | | VINS-Mono | | SenseSLAM | |
|---|---|---|---|---|---|---|---|---|---|
| APE/RPE(mm) | A0 | 156.018 | 7.436 | 71.677 | 7.064 | 63.395 | 3.510 | **58.995** | **2.525** |
| | A1 | 294.091 | 14.580 | 87.730 | 4.283 | 80.687 | 3.472 | **55.097** | **2.876** |
| | A2 | 102.657 | 10.151 | 68.381 | 5.412 | 74.842 | 8.605 | **36.370** | **1.560** |
| | A3 | 44.493 | 3.780 | 22.949 | 8.739 | 19.964 | 1.234 | **17.792** | **0.779** |
| | A4 | 114.845 | 8.338 | 146.890 | 12.460 | 18.691 | 1.091 | **15.558** | **0.930** |
| | A5 | 82.885 | 8.388 | 77.924 | 7.588 | 42.451 | 2.964 | **34.810** | **1.954** |
| | A6 | 66.001 | 6.761 | 63.895 | 6.860 | 26.240 | 1.167 | **20.467** | **0.569** |
| | A7 | 105.492 | 4.576 | 47.465 | 6.352 | 18.226 | 1.465 | **10.777** | **0.831** |
| ARE/RRE(deg) | A0 | 6.584 | 0.203 | 3.637 | 0.741 | **3.441** | 0.205 | 3.660 | **0.197** |
| | A1 | 8.703 | 0.135 | 5.140 | 1.098 | **1.518** | **0.088** | 2.676 | 0.092 |
| | A2 | 3.324 | 0.195 | 2.493 | 0.869 | 1.775 | 0.201 | **1.674** | **0.181** |
| | A3 | 6.952 | 0.186 | 2.459 | 0.825 | 2.121 | **0.176** | **1.642** | 0.182 |
| | A4 | 4.031 | 0.104 | 3.765 | 0.603 | 1.185 | **0.063** | **1.129** | 0.071 |
| | A5 | 4.928 | 0.167 | 8.843 | 0.360 | 3.000 | **0.040** | **2.041** | 0.089 |
| | A6 | 2.625 | 0.170 | 2.275 | 0.629 | **1.478** | **0.131** | 1.656 | 0.134 |
| | A7 | 6.810 | 0.120 | 3.536 | 0.602 | 1.248 | **0.073** | **0.502** | 0.082 |
| Completeness(%) | A0 | 40.186 | | 94.255 | | 92.546 | | **97.317** | |
| | A1 | 1.646 | | **98.235** | | 86.508 | | 95.072 | |
| | A2 | 61.423 | | 94.959 | | 88.301 | | **99.707** | |
| | A3 | 97.814 | | 95.972 | | **100.000** | | **100.000** | |
| | A4 | 76.629 | | 97.429 | | **100.000** | | **100.000** | |
| | A5 | 76.738 | | 98.162 | | 98.795 | | **99.143** | |
| | A6 | 94.128 | | 97.805 | | **100.000** | | **100.000** | |
| | A7 | 68.341 | | 96.690 | | **100.000** | | **100.000** | |

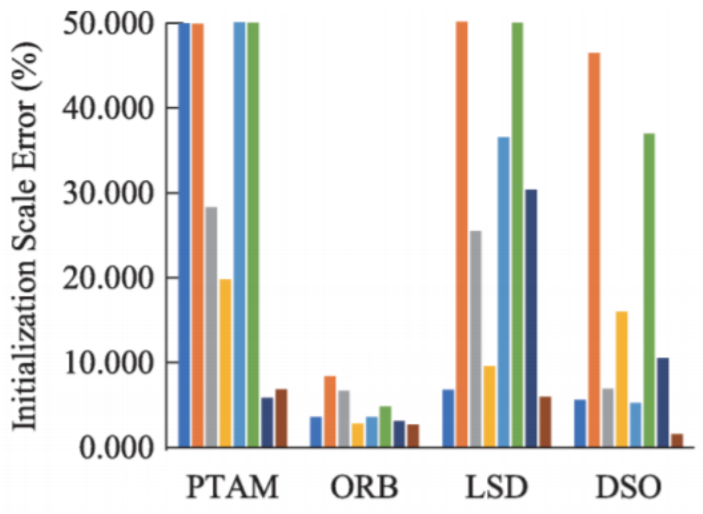
## Experimental Results

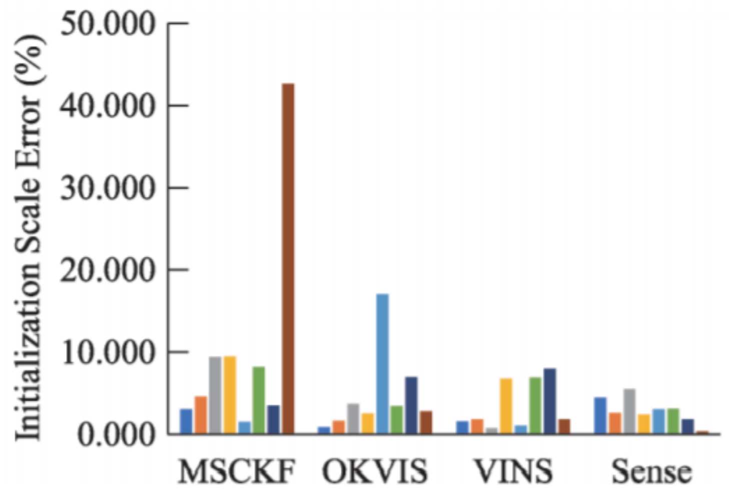
- Initialization Time





- Initialization Scale Error

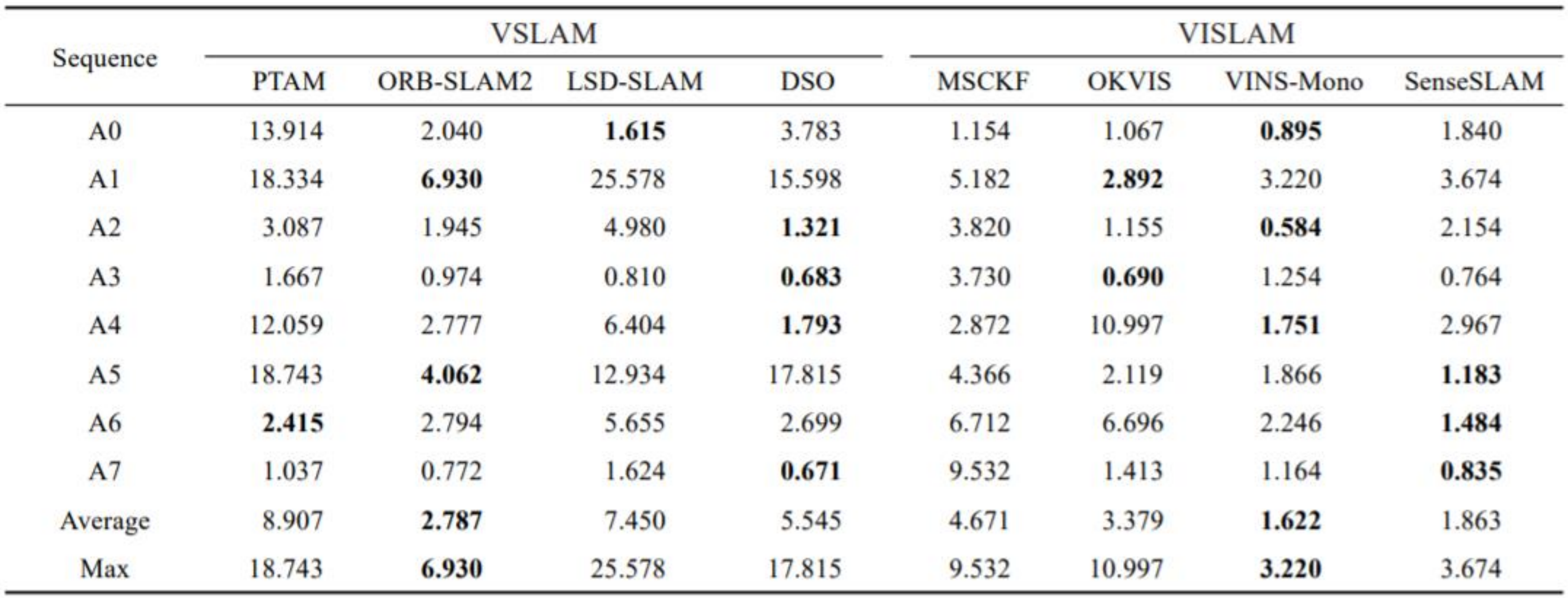




## Experimental Results

- Initialization quality

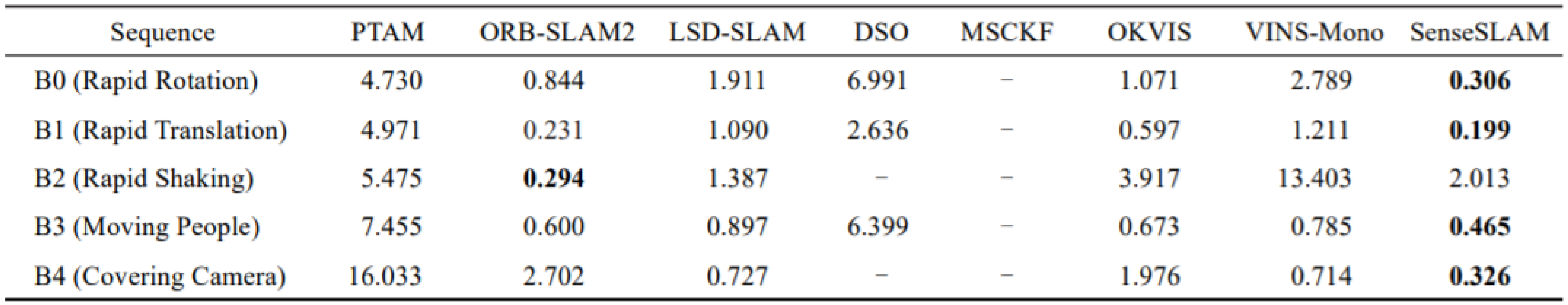
$$\epsilon_{\text{init}} = t_{\text{init}} \left( \epsilon_{\text{scale}} + \beta \right)^{\alpha}$$



| Sequence | VSLAM | | | | VISLAM | | | |
|---|---|---|---|---|---|---|---|---|
| | PTAM | ORB-SLAM2 | LSD-SLAM | DSO | MSCKF | OKVIS | VINS-Mono | SenseSLAM |
| A0 | 13.914 | 2.040 | **1.615** | 3.783 | 1.154 | 1.067 | **0.895** | 1.840 |
| A1 | 18.334 | **6.930** | 25.578 | 15.598 | 5.182 | **2.892** | 3.220 | 3.674 |
| A2 | 3.087 | 1.945 | 4.980 | **1.321** | 3.820 | 1.155 | **0.584** | 2.154 |
| A3 | 1.667 | 0.974 | 0.810 | **0.683** | 3.730 | **0.690** | 1.254 | 0.764 |
| A4 | 12.059 | 2.777 | 6.404 | **1.793** | 2.872 | 10.997 | **1.751** | 2.967 |
| A5 | 18.743 | **4.062** | 12.934 | 17.815 | 4.366 | 2.119 | 1.866 | **1.183** |
| A6 | **2.415** | 2.794 | 5.655 | 2.699 | 6.712 | 6.696 | 2.246 | **1.484** |
| A7 | 1.037 | 0.772 | 1.624 | **0.671** | 9.532 | 1.413 | 1.164 | **0.835** |
| Average | 8.907 | **2.787** | 7.450 | 5.545 | 4.671 | 3.379 | **1.622** | 1.863 |
| Max | 18.743 | **6.930** | 25.578 | 17.815 | 9.532 | 10.997 | **3.220** | 3.674 |

## Experimental Results

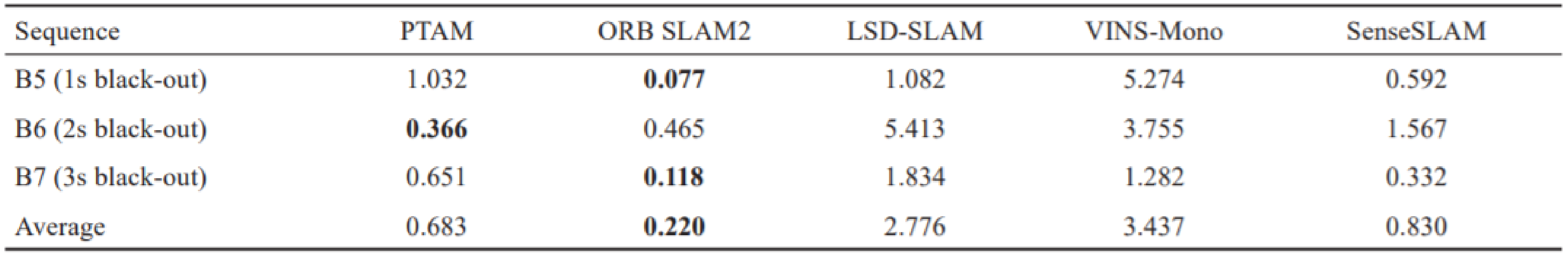
- Tracking Robustness



| Sequence | PTAM | ORB-SLAM2 | LSD-SLAM | DSO | MSCKF | OKVIS | VINS-Mono | SenseSLAM |
|---|---|---|---|---|---|---|---|---|
| B0 (Rapid Rotation) | 4.730 | 0.844 | 1.911 | 6.991 | – | 1.071 | 2.789 | **0.306** |
| B1 (Rapid Translation) | 4.971 | 0.231 | 1.090 | 2.636 | – | 0.597 | 1.211 | **0.199** |
| B2 (Rapid Shaking) | 5.475 | **0.294** | 1.387 | – | – | 3.917 | 13.403 | 2.013 |
| B3 (Moving People) | 7.455 | 0.600 | 0.897 | 6.399 | – | 0.673 | 0.785 | **0.465** |
| B4 (Covering Camera) | 16.033 | 2.702 | 0.727 | – | – | 1.976 | 0.714 | **0.326** |

## Experimental Results

- Relocalization Time



| Sequence | PTAM | ORB SLAM2 | LSD-SLAM | VINS-Mono | SenseSLAM |
|---|---|---|---|---|---|
| B5 (1s black-out) | 1.032 | **0.077** | 1.082 | 5.274 | 0.592 |
| B6 (2s black-out) | **0.366** | 0.465 | 5.413 | 3.755 | 1.567 |
| B7 (3s black-out) | 0.651 | **0.118** | 1.834 | 1.282 | 0.332 |
| Average | 0.683 | **0.220** | 2.776 | 3.437 | 0.830 |

Benchmark website: http://www.zjucvg.net/eval-vislam/

Jinyu Li, Bangbang Yang, Danpeng Chen, Nan Wang, Guofeng Zhang*, Hujun Bao*. Survey and Evaluation of Monocular Visual-Inertial SLAM Algorithms for Augmented Reality. Journal of Virtual Reality & Intelligent Hardware, published online: http://www.vr-ih.com/vrih/html/EN/10.3724/SP.J.2096-5796.2018.0011/article.html

## Online Side-by-Side Comparison

### Problem:

**Most commercial software does not support loading sequences mode.**

### Side-by-Side comparison:

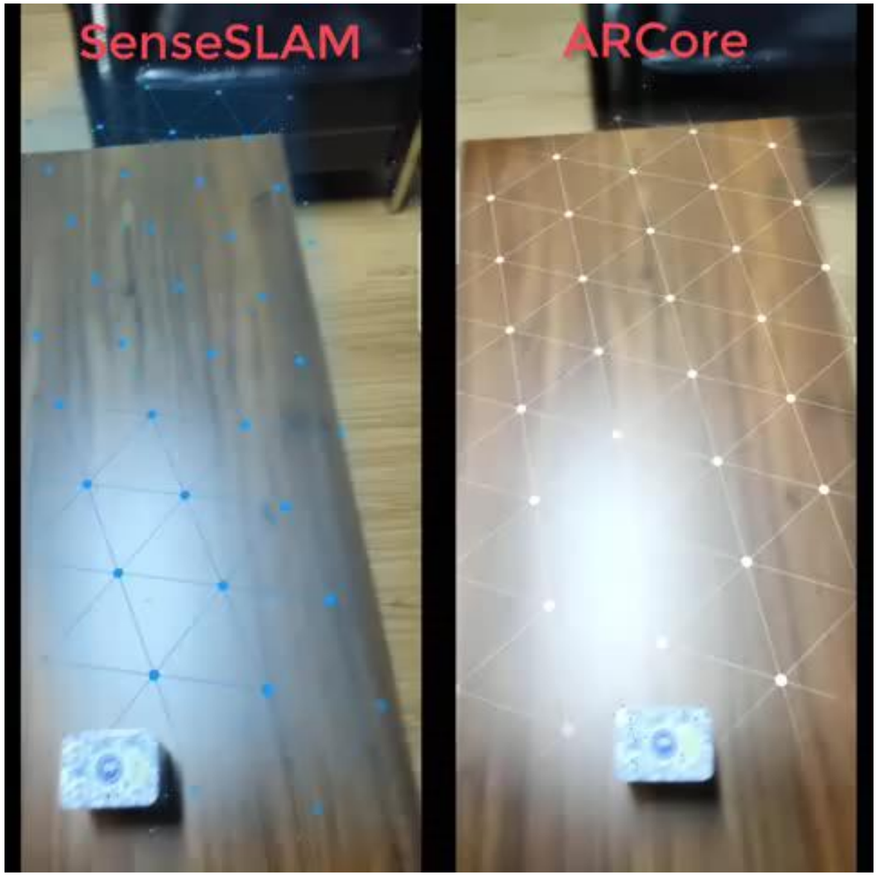
- ☐ Online real-time running
- ☐ Simultaneously capture trajectory
- ☐ Trajectory alignment and accuracy evaluation
- ☐ 1vs1 for all the criteria

# Comparisons

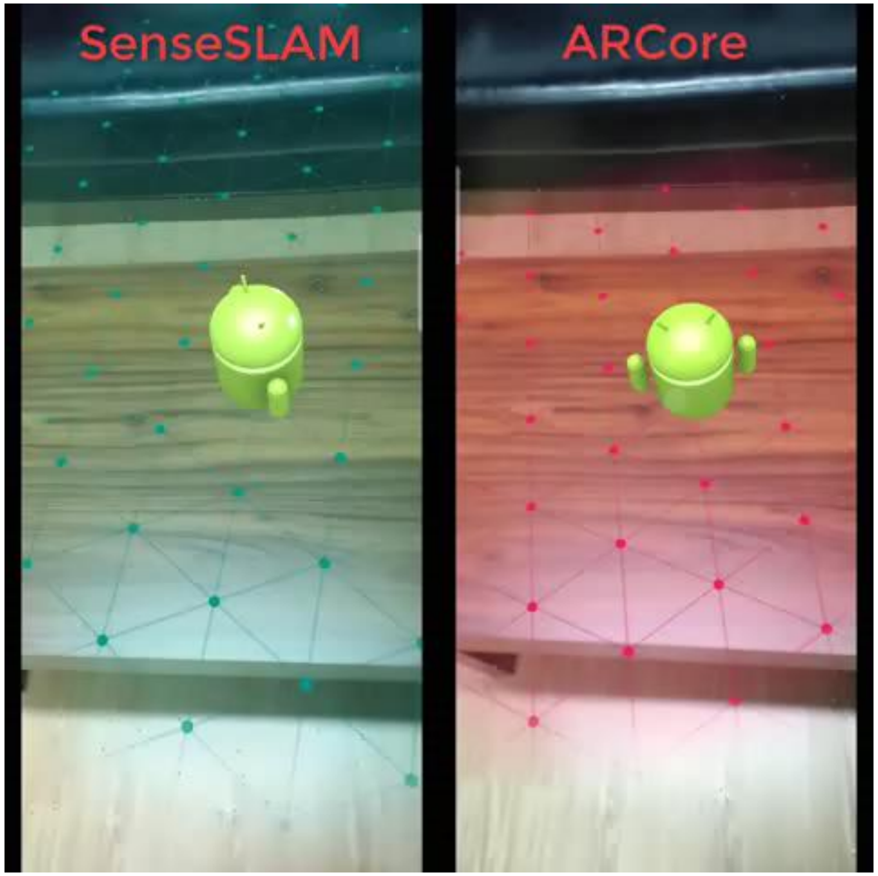
| System | Scene | APE[mm] | Completeness (%) | Initialization Quality | Tracking Robustness |
|---|---|---|---|---|---|
| ARCore v1.9 | 01 | 67.498 | 88.882 | 2.985055 | 0.337 |
| SenseSLAM v2.0 | 01 | 73.735 | 91.593 | 1.781471 | 0.369 |
| ARCore v1.9 | 02 | 76.302 | 93.072 | 2.465614 | 0.382 |
| SenseSLAM v2.0 | 02 | 100.999 | 56.631 | 2.238261 | 0.505 |
| ARCore v1.9 | 03 | 179.567 | 40.224 | 0.284811 | 6.064 |
| SenseSLAM v2.0 | 03 | 52.895 | 94.591 | 3.204471 | 0.264 |
| ARCore v1.9 | 04 | 134.114 | 44.563 | 5.421213 | 0.671 |
| SenseSLAM v2.0 | 04 | 122.414 | 47.254 | 2.697703 | 0.612 |
| ARCore v1.9 | 05 | 76.461 | 91.667 | 1.059667 | 0.382 |
| SenseSLAM v2.0 | 05 | 71.024 | 89.482 | 2.381186 | 0.355 |
| ARCore v1.9 | 06 | 85.098 | 82.042 | 1.36916 | 0.624 |
| SenseSLAM v2.0 | 06 | 86.276 | 72.702 | 3.577339 | 0.431 |
| ARCore v1.9 | 07 | 112.105 | 56.695 | 1.304656 | 0.561 |
| SenseSLAM v2.0 | 07 | 74.66 | 81.979 | 1.221794 | 0.373 |

# AR Performance Comparison

**Robustness and relocalization efficiency under occlusion**



**Robustness and relocalization efficiency under fast motion**

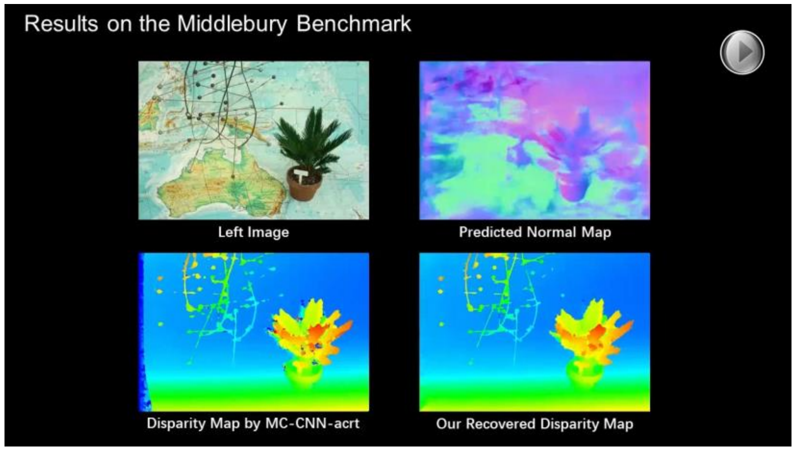


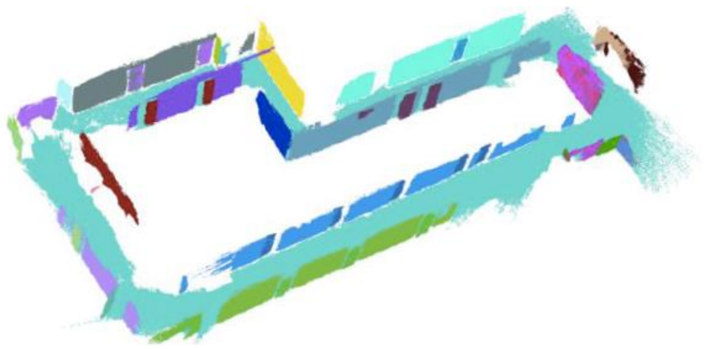
# ZJU3DV开源开放计划

- 已开源或开放的SLAM系统、算法或数据集
  - RDSLAM
    - http://www.zjucvg.net/rdslam/rdslam.html
  - RKSLAM
    - http://www.zjucvg.net/rkslam/rkslam.html
  - SenseSLAM v1.0
    - http://www.zjucvg.net/senseslam/
  - EIBA：https://github.com/zju3dv/EIBA
  - ZJU - SenseTime VISLAM Benchmark
    - http://www.zjucvg.net/eval-vislam/
- **未来将开源开放更多SLAM算法/系统或数据集**
  - http://www.zjucvg.net
  - http://github.com/zju3dv

## Visual SLAM技术发展趋势（1）

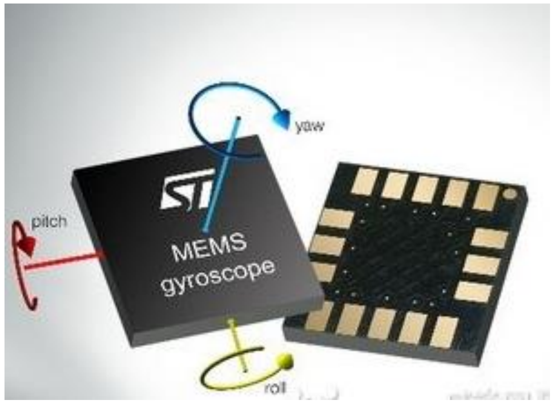
- 缓解特征依赖
  - 基于边的跟踪
  - 直接图像跟踪或半稠密跟踪
  - 结合机器学习和先验/语义信息
- 稠密三维重建
  - 单 / 多目实时三维重建
  - 基于深度相机的实时三维重建
  - 平面表达和模型自适应简化


Results on the Middlebury Benchmark
Left Image
Predicted Normal Map
Disparity Map by MC-CNN-acrt
Our Recovered Disparity Map



## Visual SLAM技术发展趋势（2）

- 多传感器融合
  - 结合IMU、GPS、深度相机、光流计、里程计



谢谢！