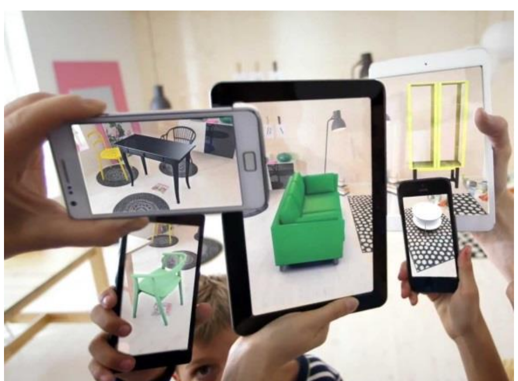
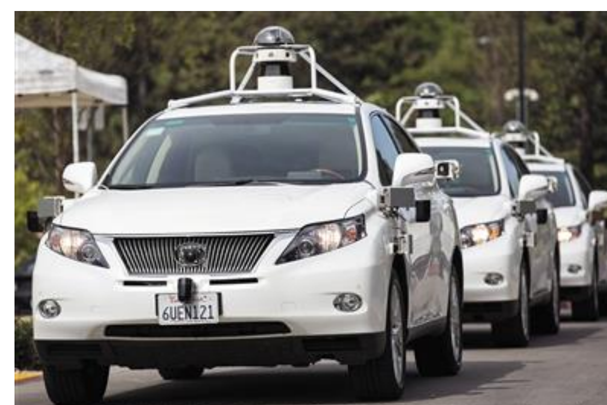
视觉SLAM

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。

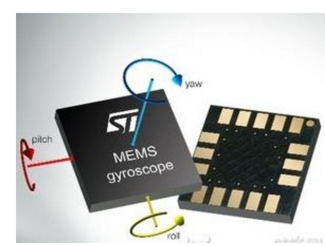
激光雷达

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

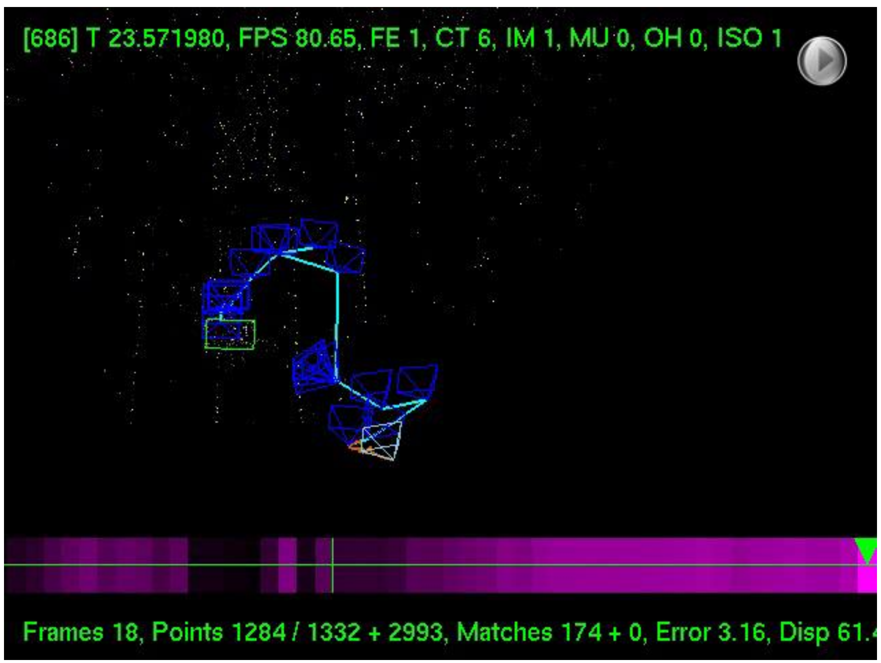
微软Kinect彩色-深度（RGBD）传感器



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）

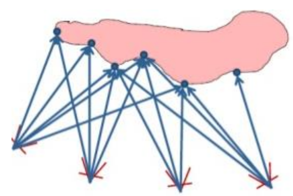


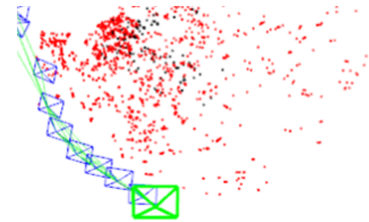
稀疏SLAM

稠密SLAM

SLAM系统常用的框架

RGB图 深度图 IMU测量值





输入

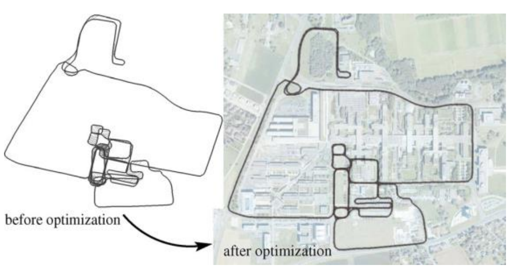
- 传感器数据

前台线程

- 根据传感器数据进行跟踪求解，实时恢复每个时刻的位姿

输出

- 设备实时位姿
- 三维点云



优化以减少误差累积

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

回路检测

Related Work

- 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

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

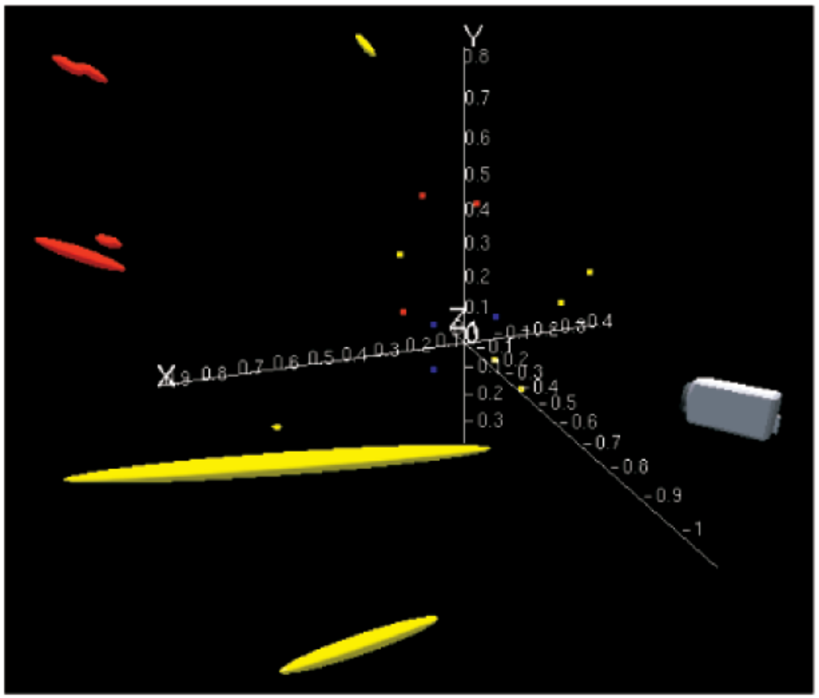
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - h(\hat{x}_{k|k-1}))$$

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

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

MonoSLAM

- Map representation



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

camera state

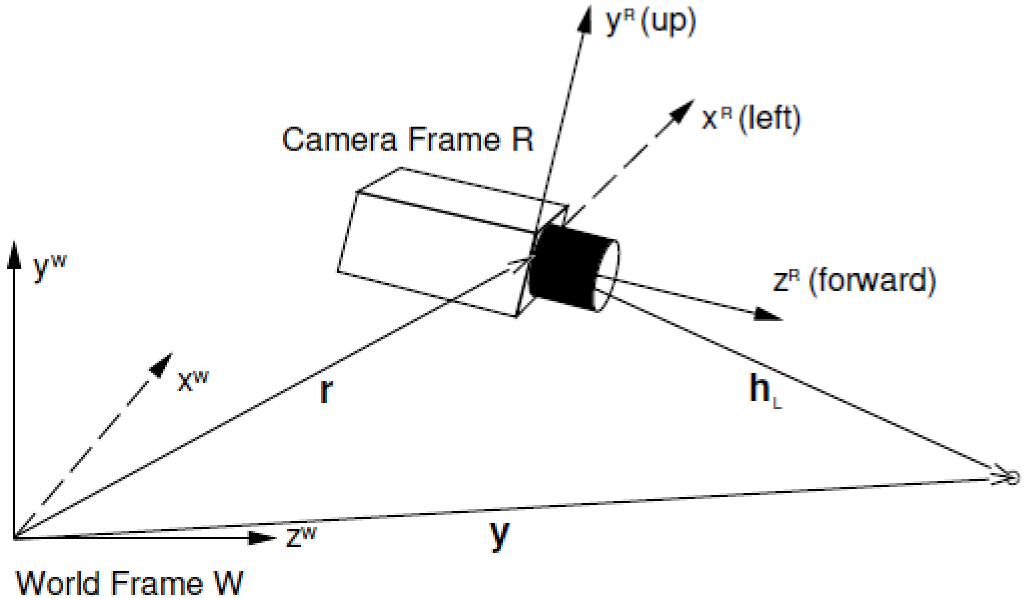
point state

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

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

$$w_k = \begin{pmatrix} a_k \\ \alpha_k \end{pmatrix} \quad \begin{matrix} \text{linear acceleration} \\ \text{angular acceleration} \end{matrix}$$

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

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

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

MonoSLAM

- Predicted features position

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

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

- Innovation covariance
  - Elliptical feature search region

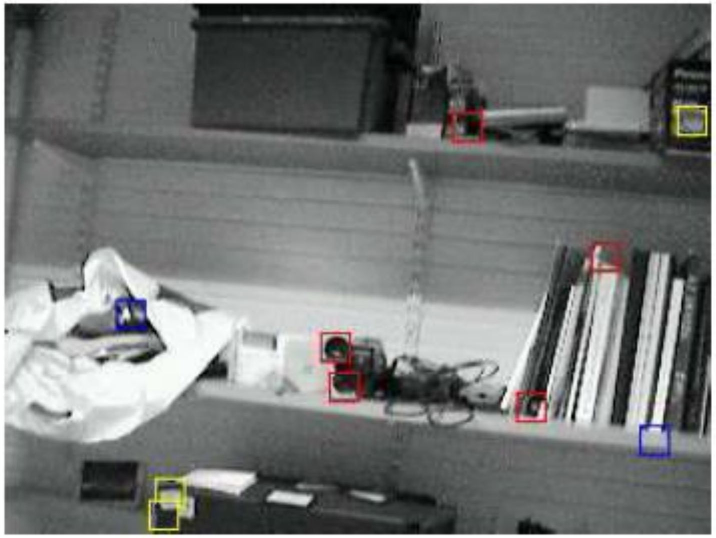
$$S_i = J_C P_{CC} J_C^T + J_C P_{CX_i} J_{X_i}^T + J_{X_i} P_{X_i C} J_C^T + J_{X_i} P_{X_i X_i} J_{X_i}^T + R$$

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

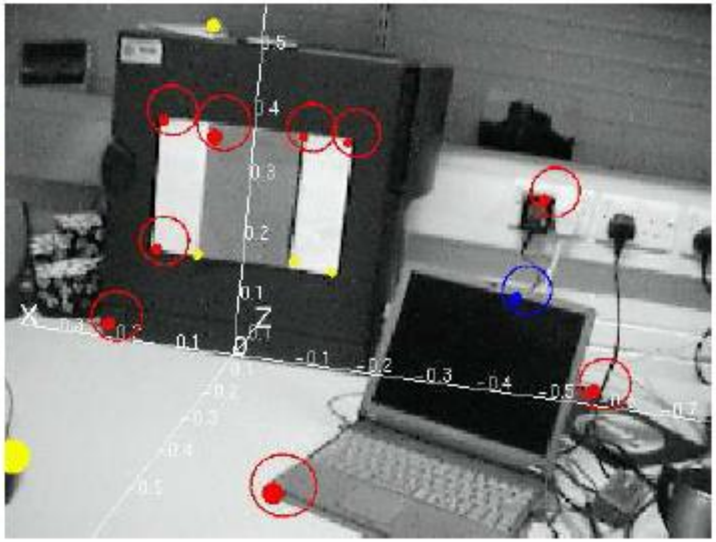
$$J_{X_i} = \frac{\partial z_i}{\partial X_i}$$

MonoSLAM

- Active search



Shi and Tomasi Feature



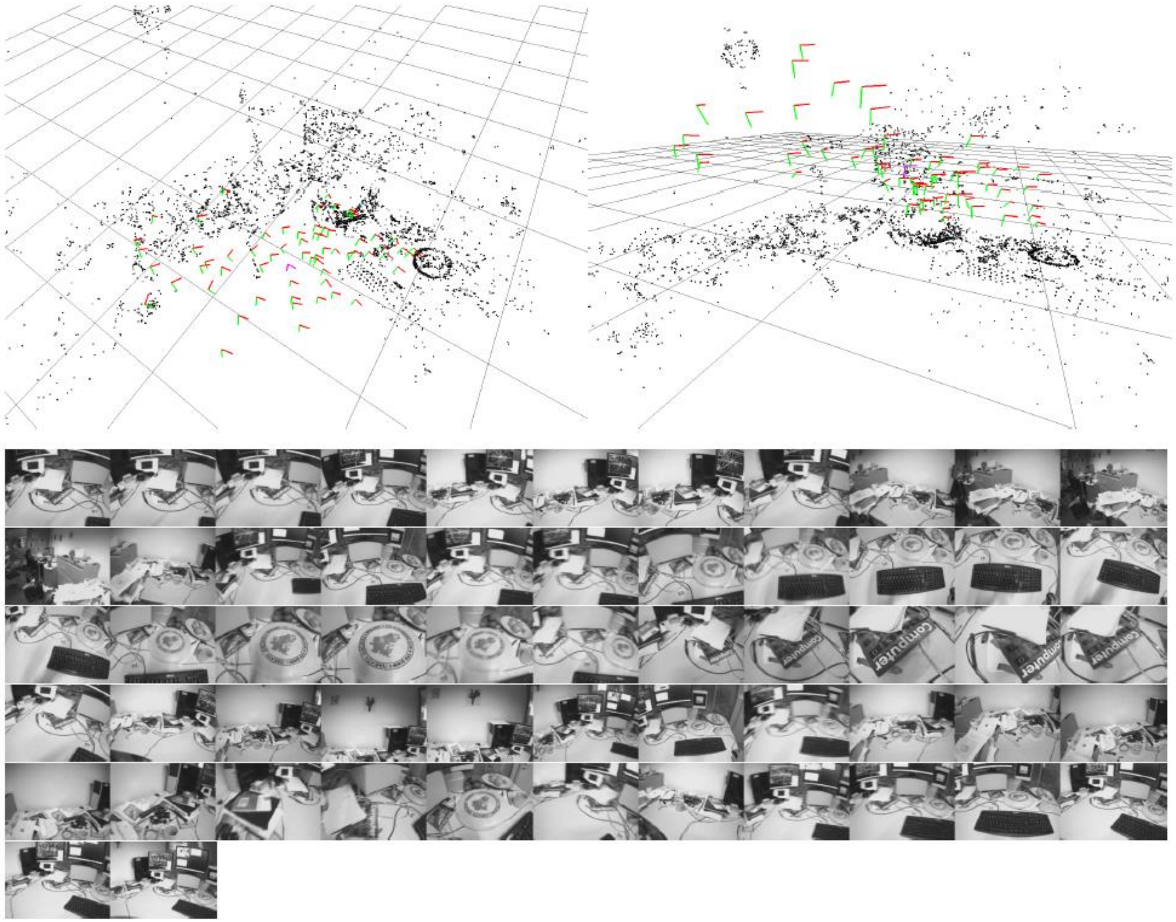
Elliptical search region

MonoSLAM

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

PTAM: Parallel Tracking and Mapping

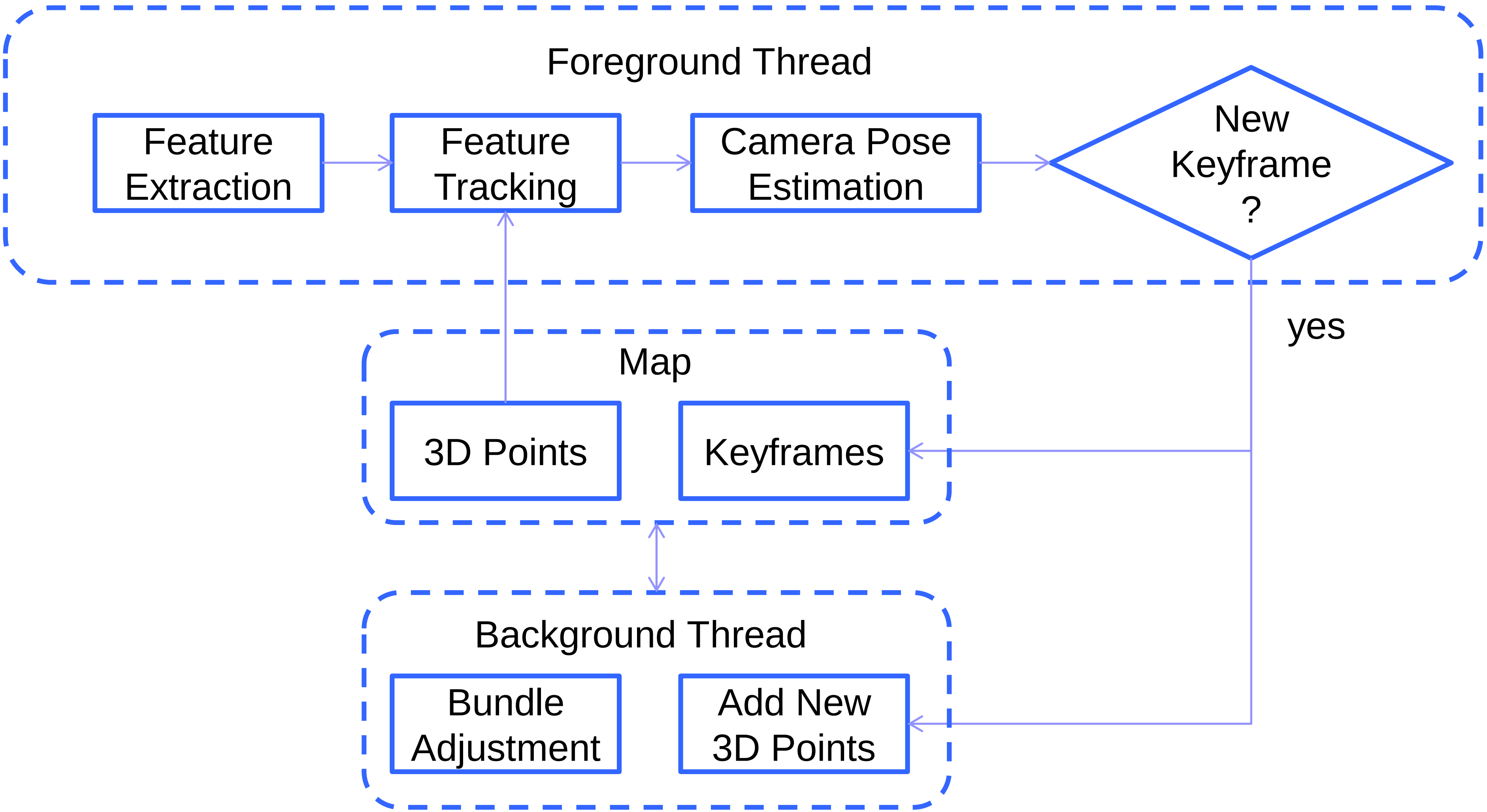
- Map representation



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.

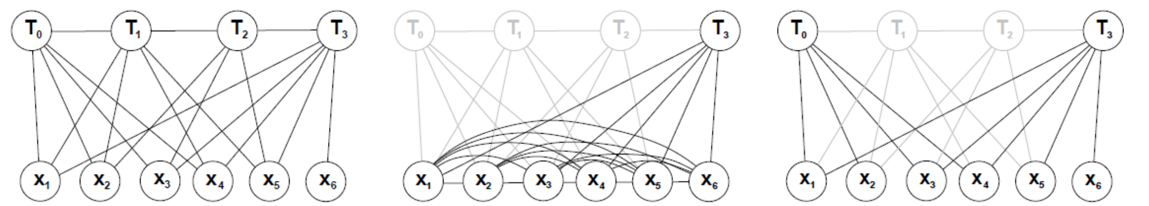
PTAM: Parallel Tracking and Mapping

- Overview



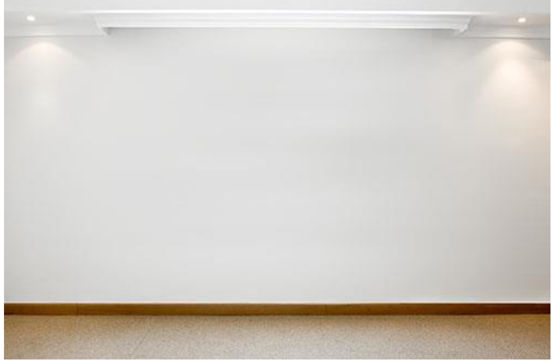
Keyframe-based SLAM vs Filtering-based SLAM

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

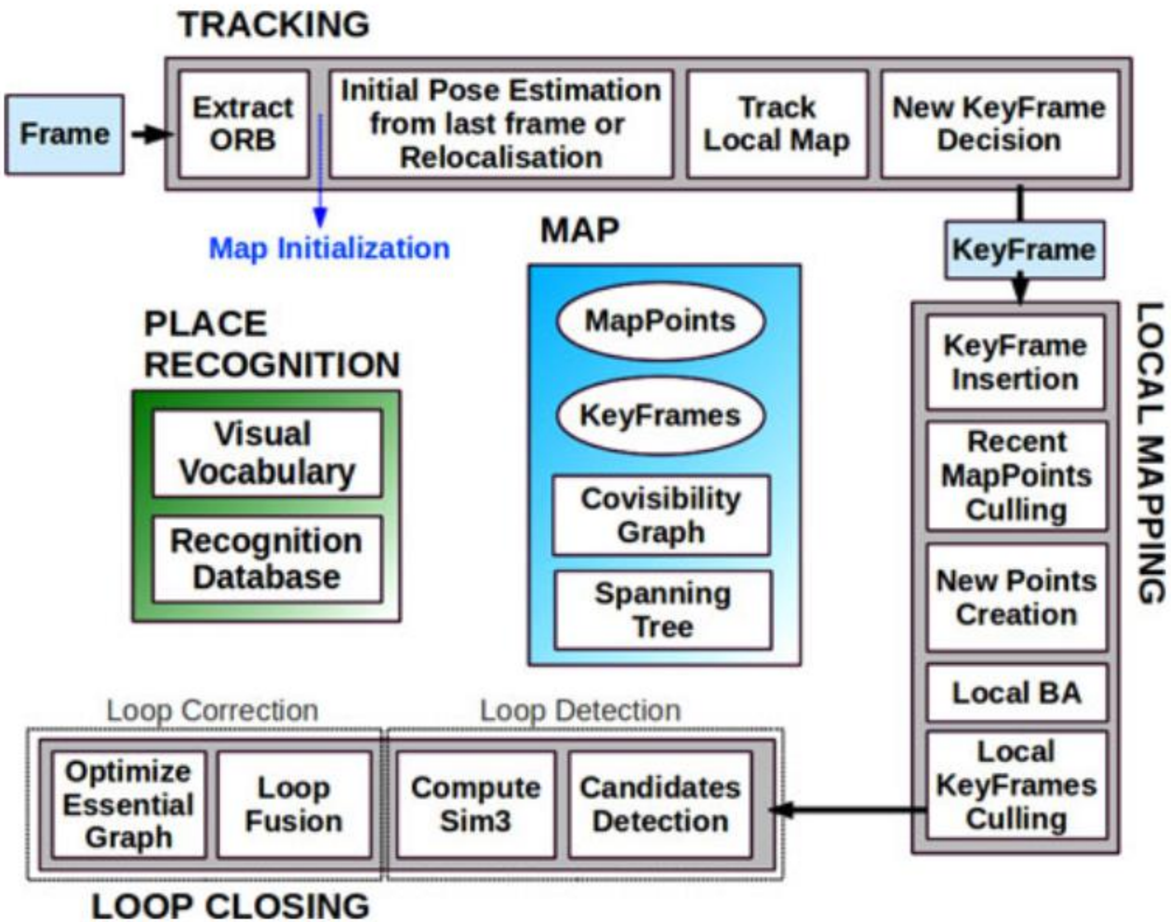


(a) Markov Random Field (b) Filter (c) Keyframe BA

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



ORB-SLAM

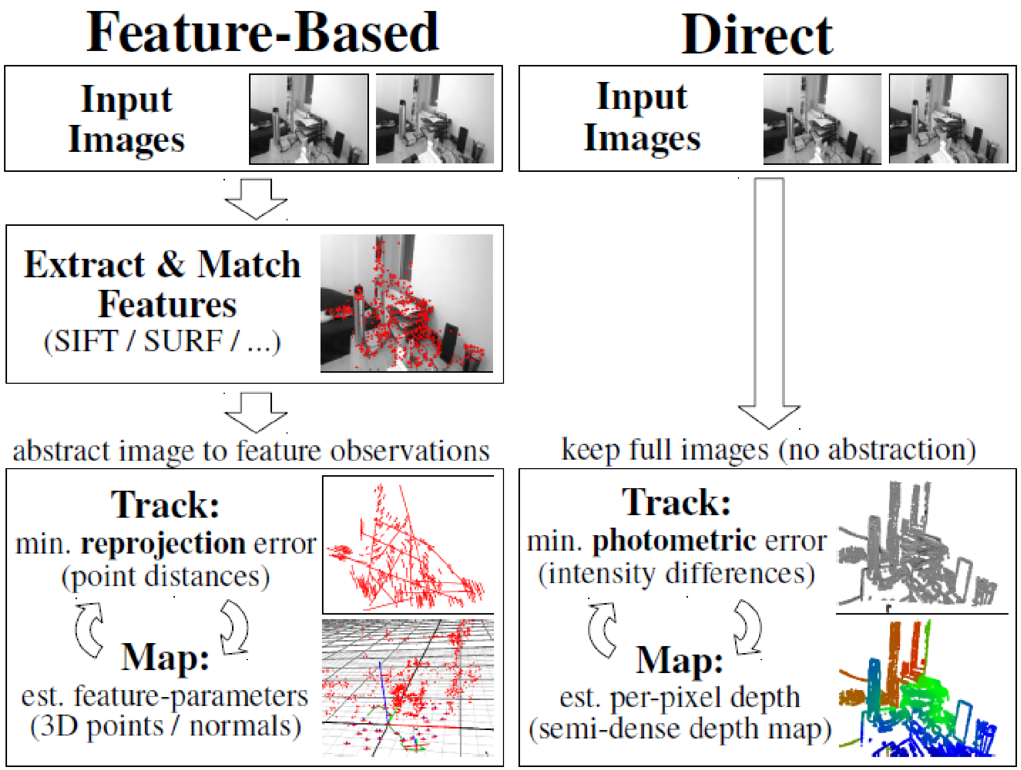


Raul Mur-Artal, J. M. M. Montiel, Juan D. Tardós: ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Trans. Robotics 31(5): 1147-1163 (2015).

ORB-SLAM: A Versatile and Accurate Monocular SLAM System

- 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 ξ by aligning intensity images I_1 and I_2 with depth map Z_1 of I_1
- Assumption

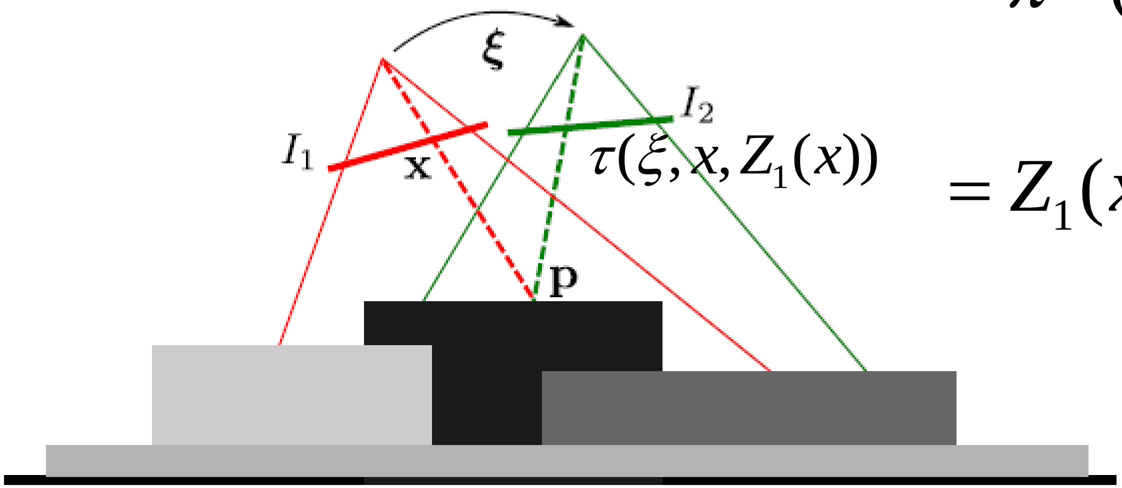
$$I_1(x) = I_2(\tau(\xi, x, Z_1(x)))$$

warping function: maps a pixel from I_1 to I_2

Direct Tracking

- Warping function

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



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

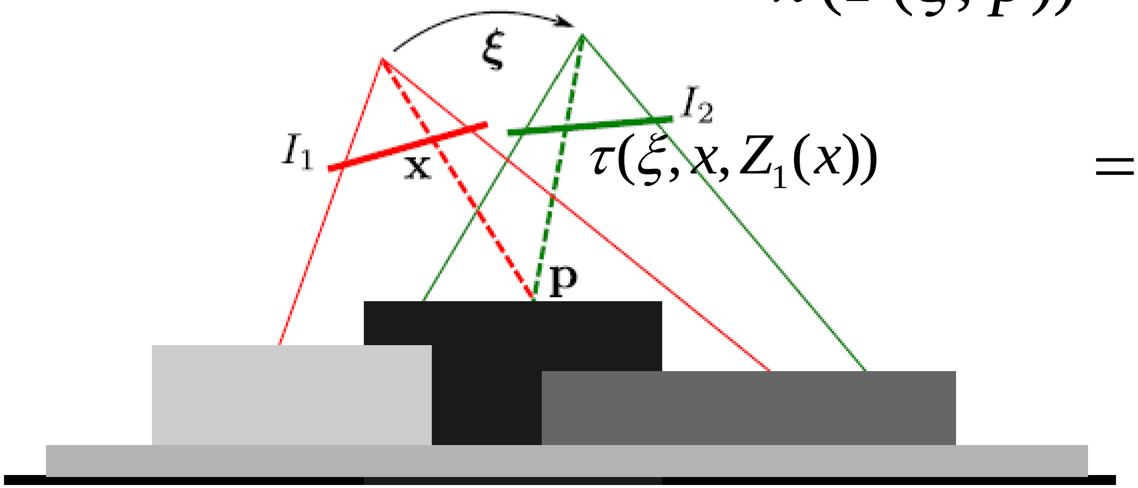
Direct Tracking

- Warping function

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

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

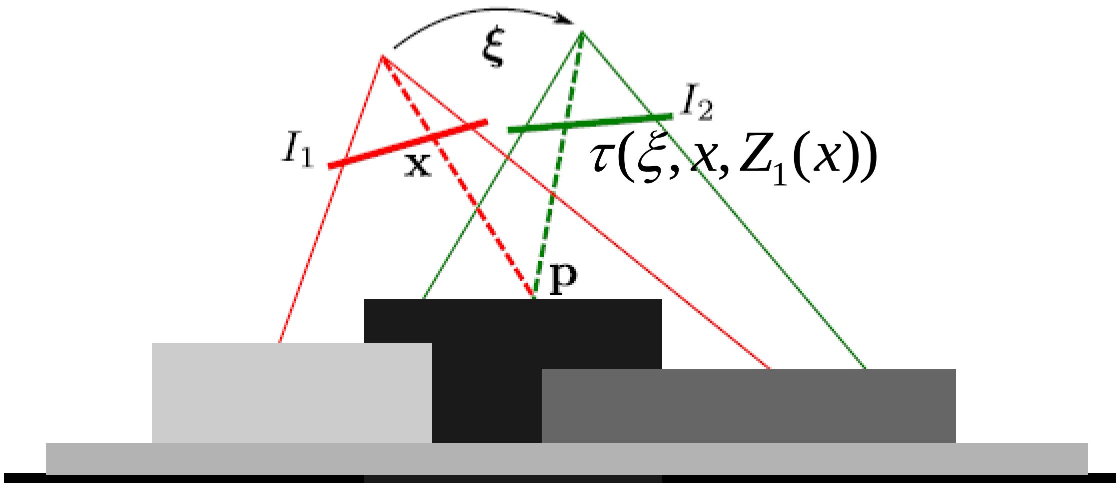
$$= \left(\frac{f_x X}{Z} + c_x, \frac{f_y Y}{Z} + c_y \right)^T$$



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

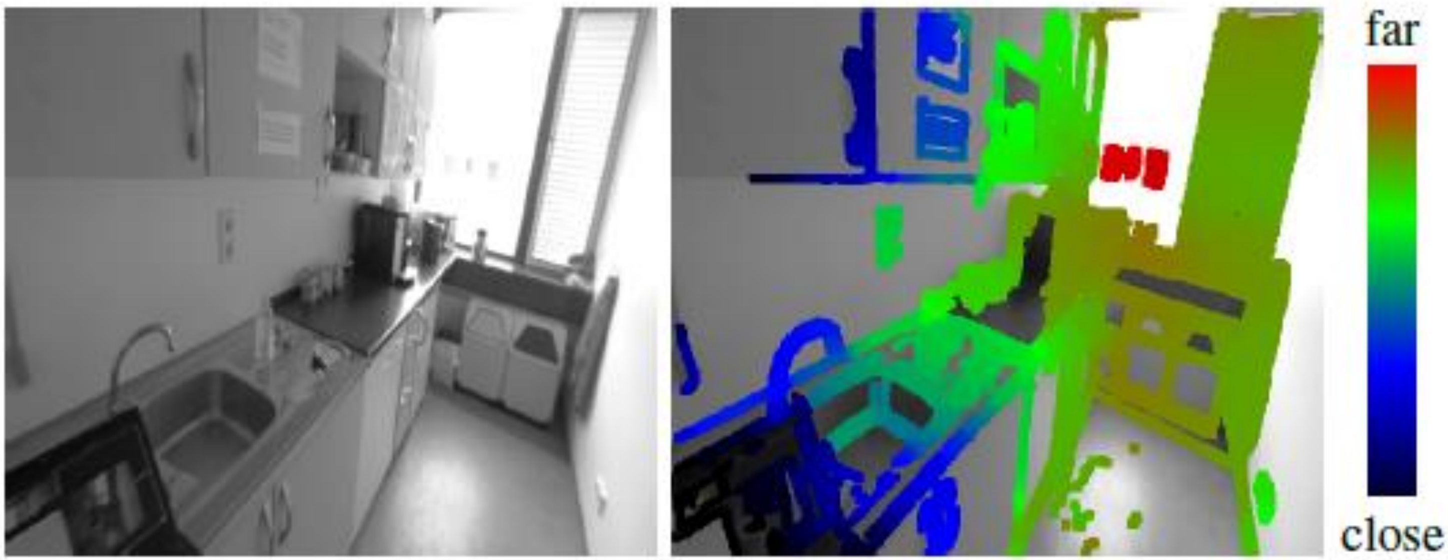
- Residual of the *k*-th pixel

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

- 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

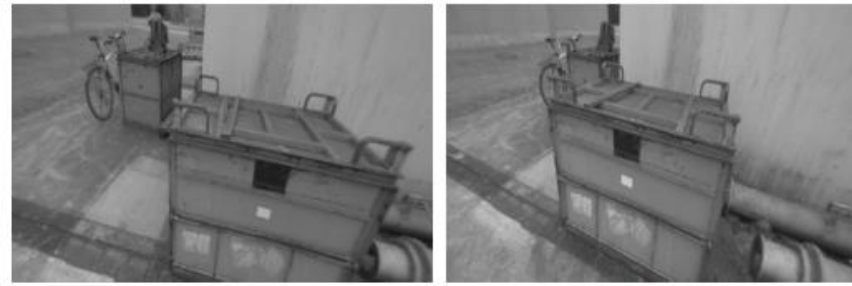
- Keyframe representation

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

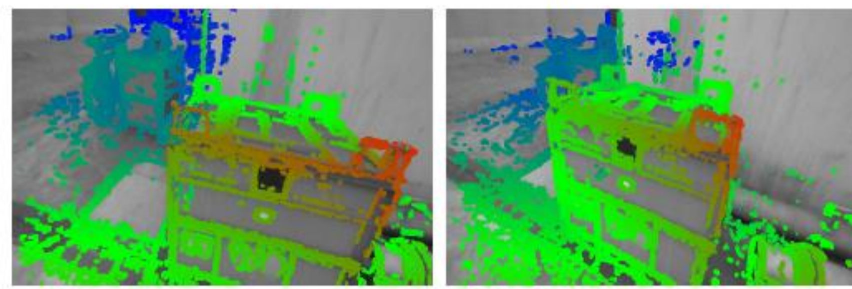
$i_i = I_i(x)$ image intensity

$d_i = D_i(x)$ inverse depth

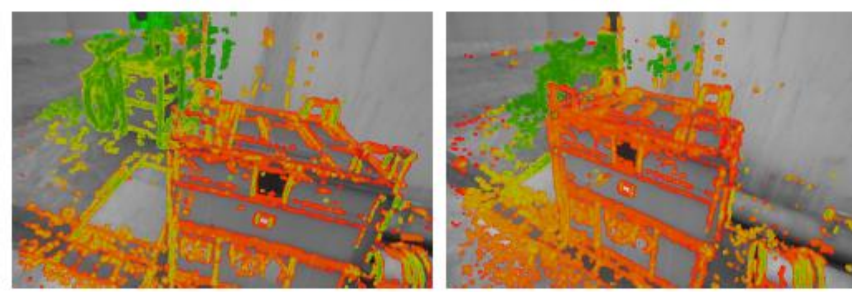
$\sigma_{d_i}^2 = V_i(x)$ inverse depth variance



(a) camera images I



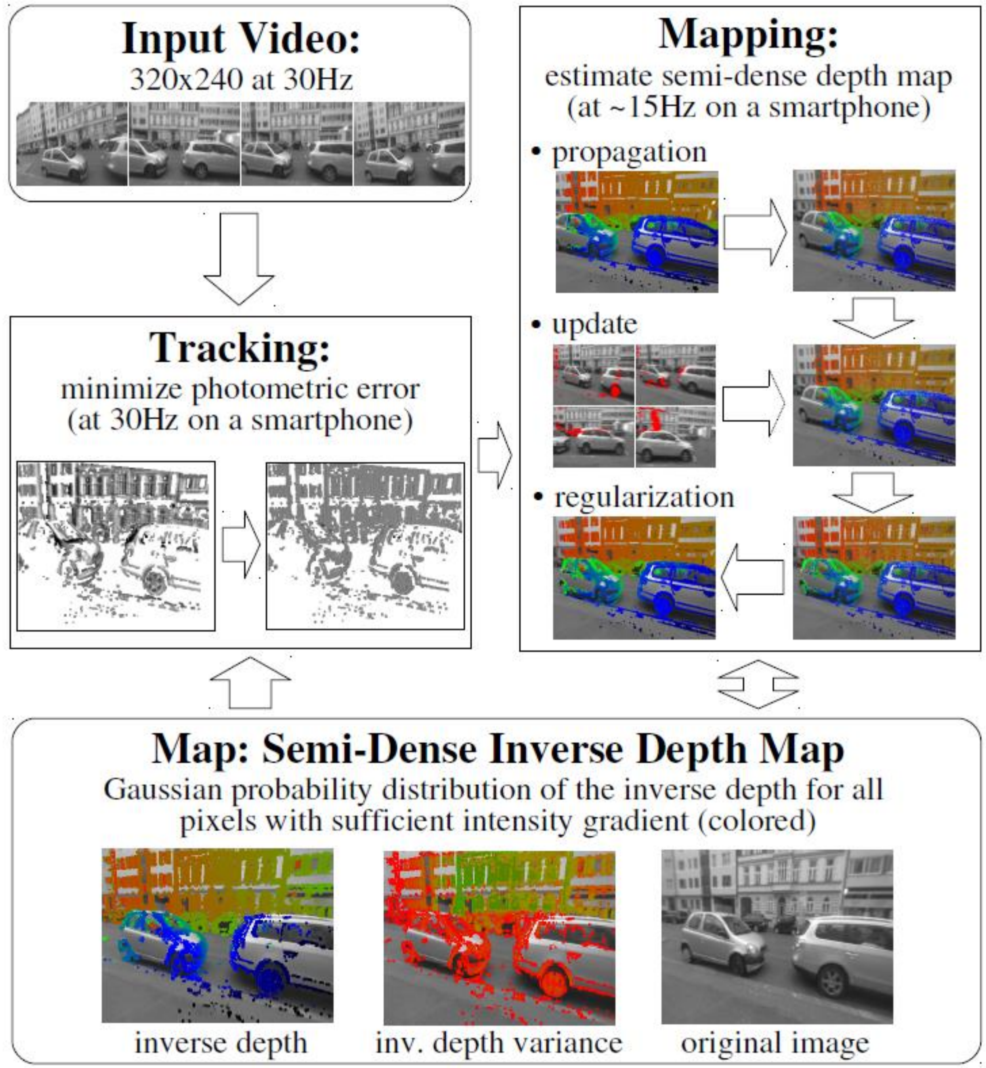
(b) estimated inverse depth maps D



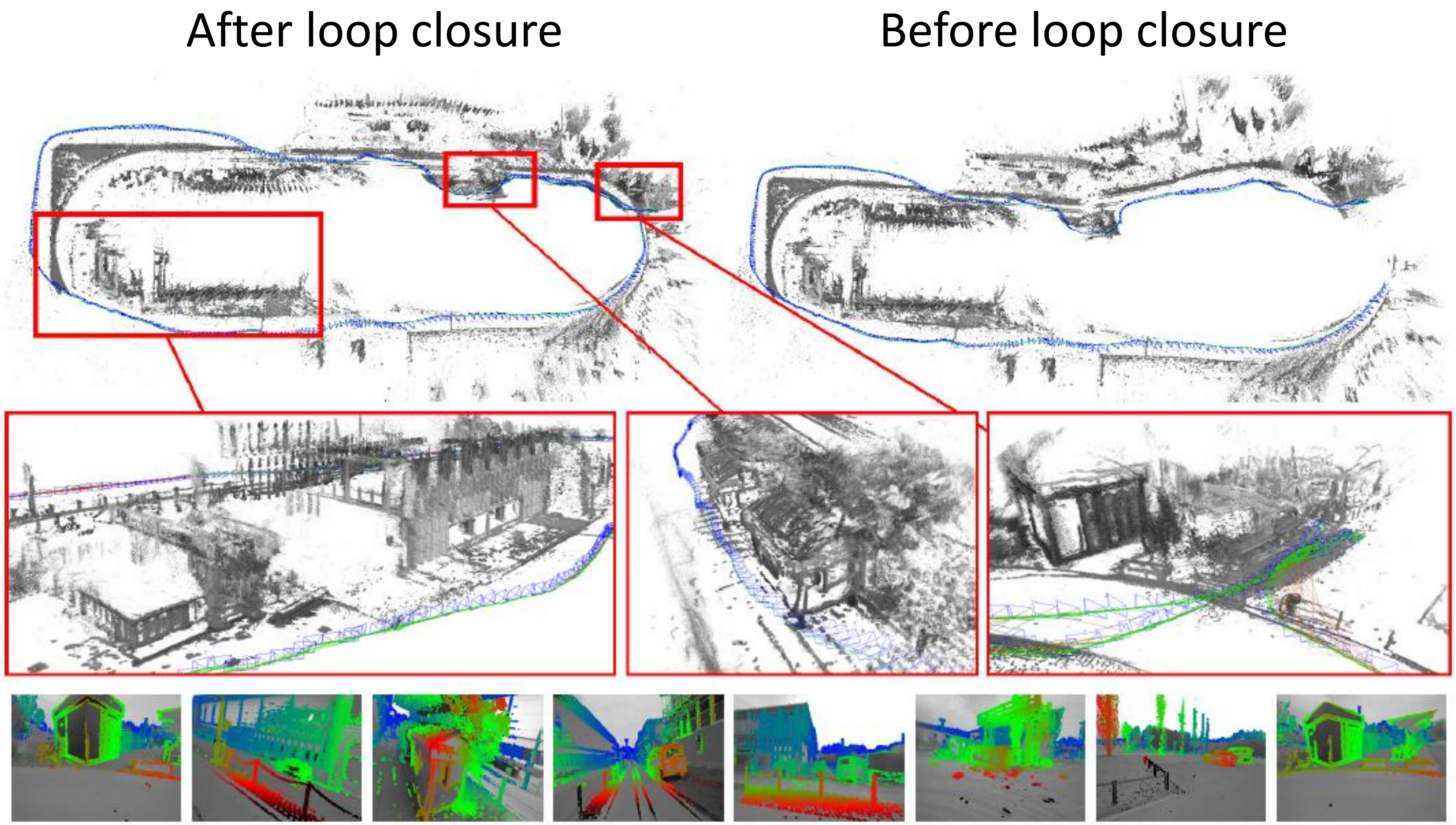
(c) inverse depth variance V

Semi-Dense Visual Odometry

- Overview



LSD-SLAM



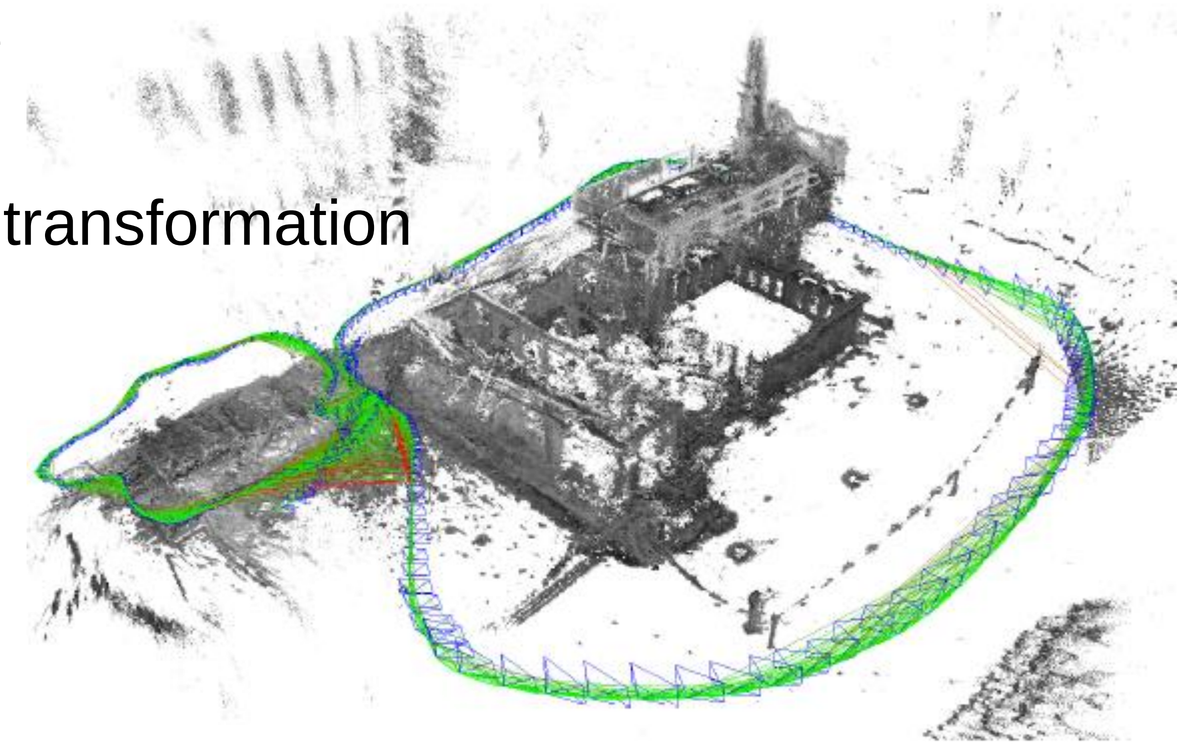
Jakob Engel, Thomas Schops, Daniel Cremers: LSD-SLAM: Large-Scale Direct Monocular SLAM. ECCV (2) 2014: 834-849.

LSD-SLAM

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

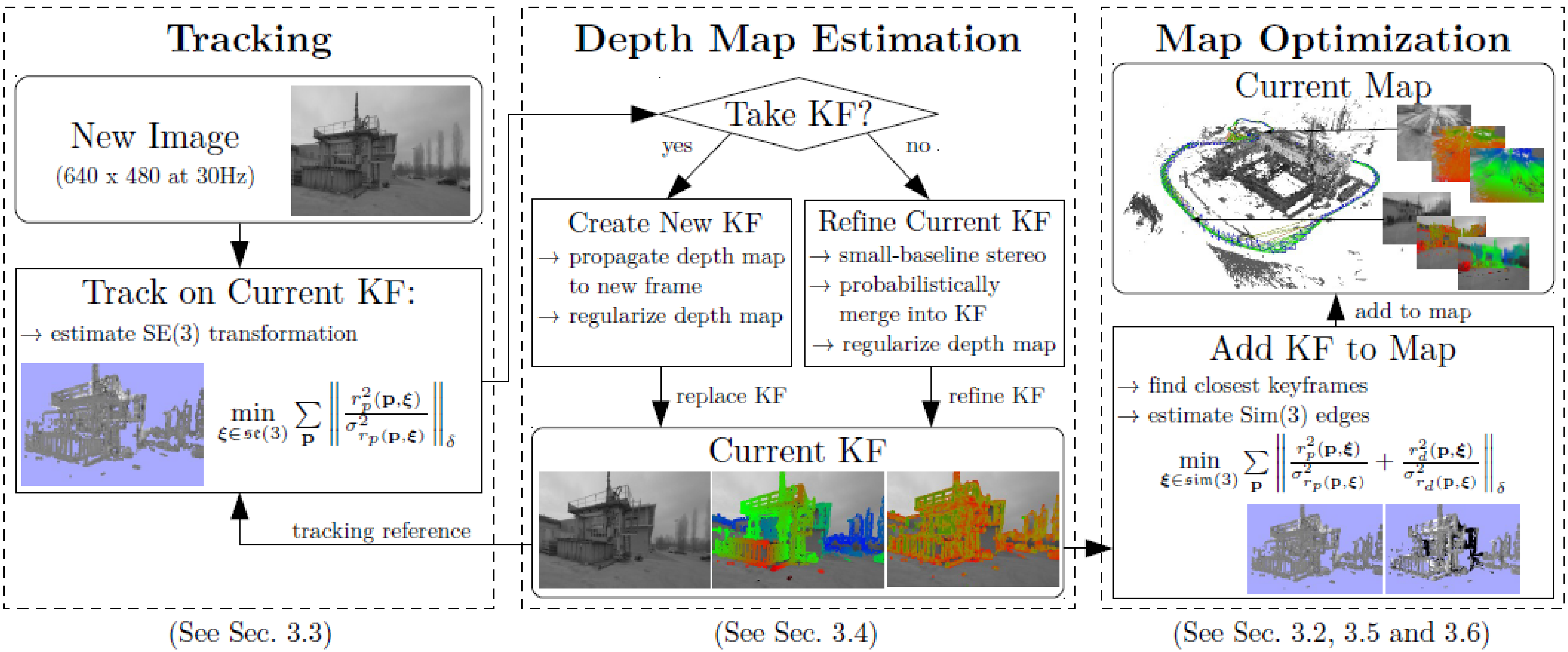
    $K_i = (I_i, D_i, V_i)$
  - Edge: similarity transformation

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



LSD-SLAM

- Overview



LSD-SLAM

- Direct sim(3) image alignment

$$\xi_{ji}^{*} = \arg\min_{\xi_{ji}} \sum_{p} \left\| \frac{r_p^2(p,\xi_{ji})}{\sigma^2_{r_p^2(p,\xi_{ji})}} + \frac{r_d^2(p,\xi_{ji})}{\sigma^2_{r_d^2(p,\xi_{ji})}} \right\|_{\delta}$$

$$r_p(p,\xi_{ji}) = I_j(\tau(\xi_{ji}, p, 1/d_i)) - I_i(p)$$

$$\sigma^2_{r_p^2(p,\xi_{ji})} = 2\sigma_I^2 + \left(\frac{\partial r_p}{\partial d_i}\right)^2 \sigma_{d_i}^2$$

$$r_d(p,\xi_{ji}) = 1/T_Z(\xi_{ji}, \pi^{-1}(p, 1/d_i)) - D_j(p_\tau)$$

$$\sigma^2_{r_d^2(p,\xi_{ji})} = V_j(p_\tau)\left(\frac{\partial r_d}{D_j(p_\tau)}\right)^2 + V_i(p)\left(\frac{\partial r_d}{D_i(p)}\right)^2$$

$$p_\tau = \tau(\xi_{ji}, p, 1/d_i)$$

LSD-SLAM

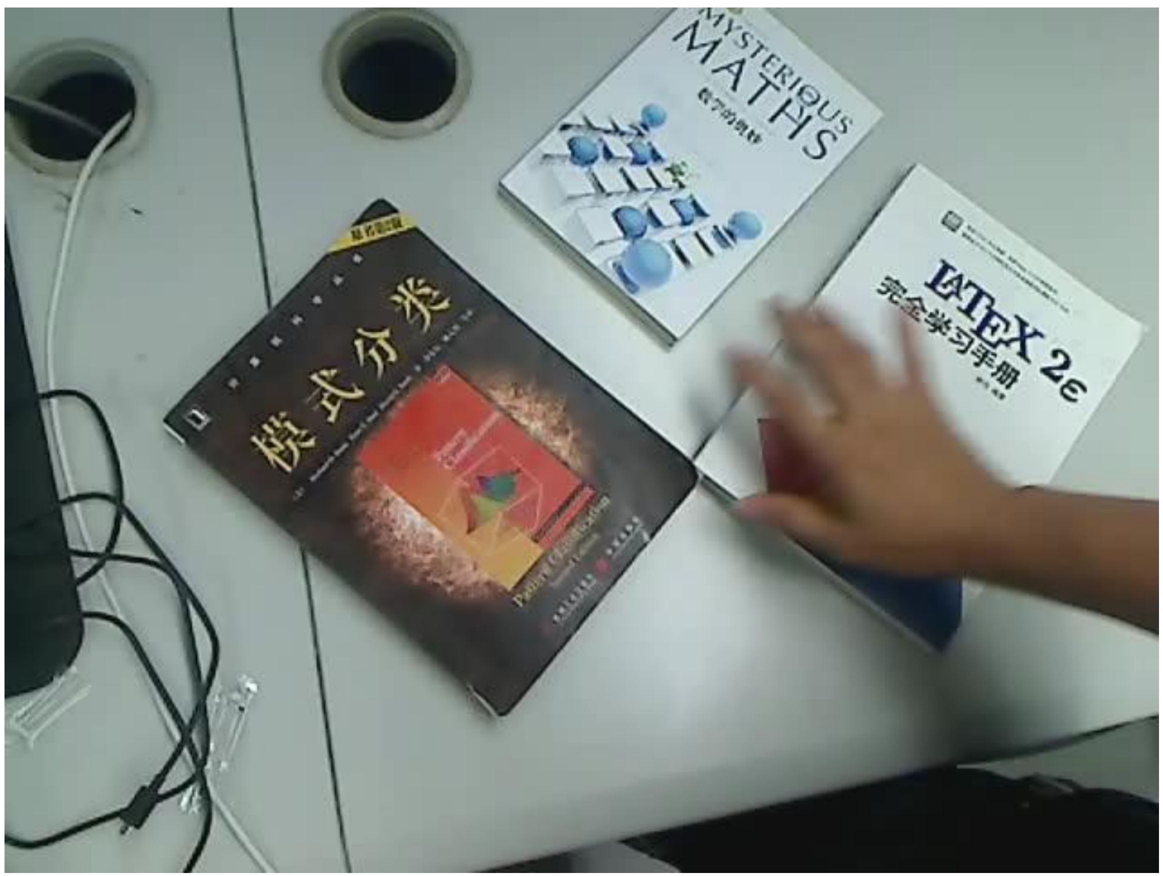
- Pose graph optimization
  - Energy function:

$$E(\xi_{W1} \dots \xi_{Wn}) := \sum_{(\boldsymbol{\xi}_{ji}, \boldsymbol{\Sigma}_{ji}) \in \mathcal{E}} (\boldsymbol{\xi}_{ji} \circ \boldsymbol{\xi}_{Wi}^{-1} \circ \boldsymbol{\xi}_{Wj})^T \boldsymbol{\Sigma}_{ji}^{-1} (\boldsymbol{\xi}_{ji} \circ \boldsymbol{\xi}_{Wi}^{-1} \circ \boldsymbol{\xi}_{Wj})$$

Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g2o: A general framework for graph optimization. In: Intl. Conf. on Robotics and Automation(ICRA) (2011)

Key Issues for SLAM in Dynamic Environments

- Gradually changing



Key Issues for SLAM in Dynamic Environments

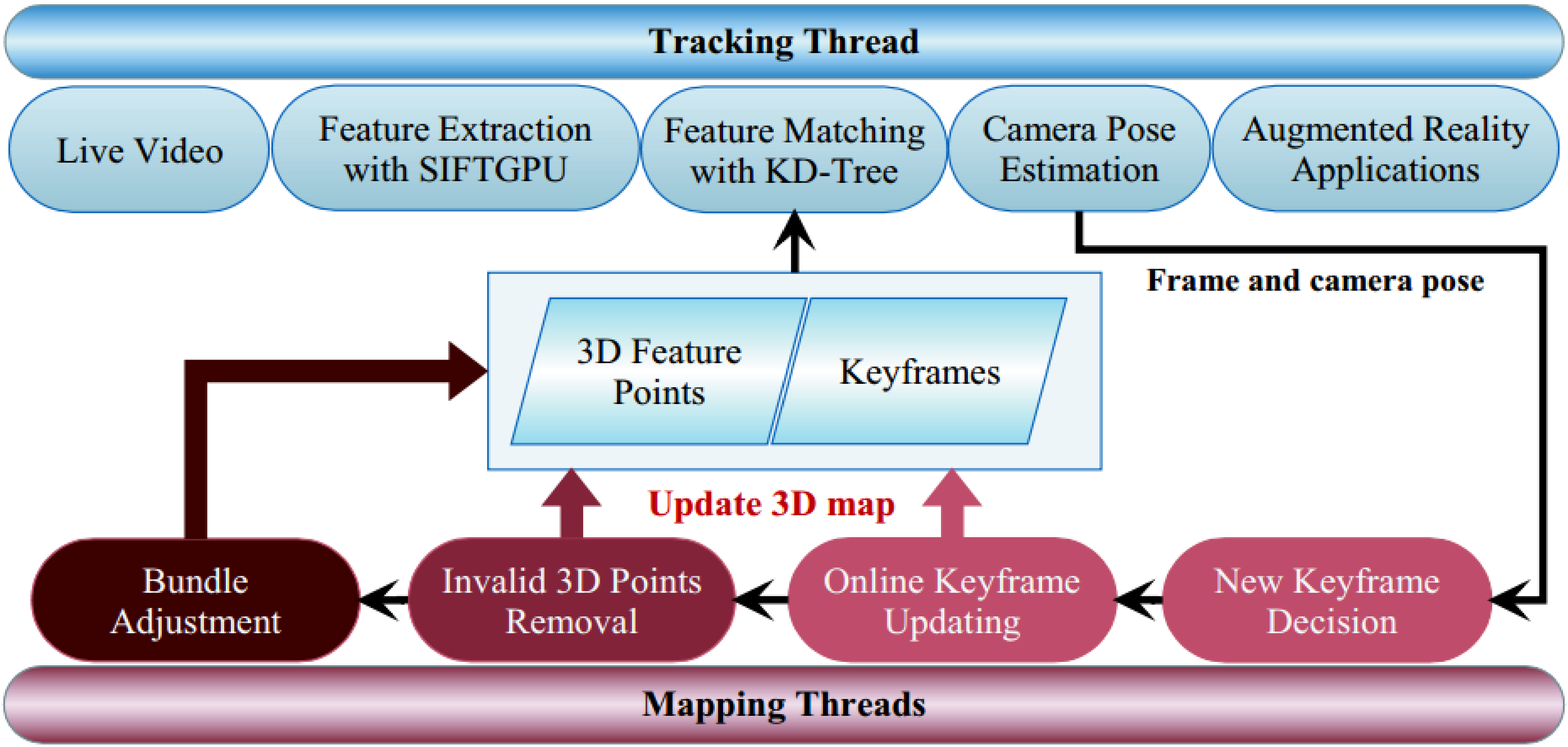
- Gradually changing

- Object Occlusion
  - Viewpoint Change
  - Dynamic Objects

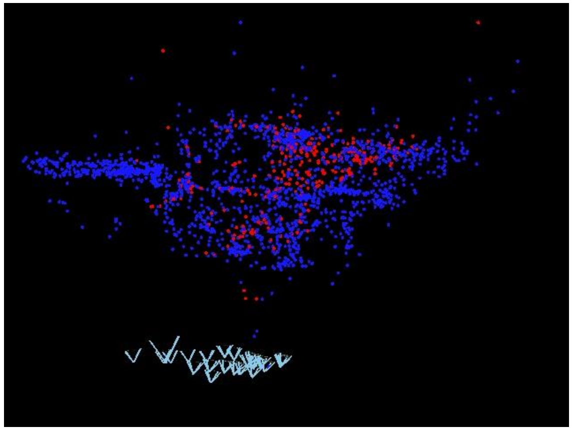
Key Issues for SLAM in Dynamic Environments

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

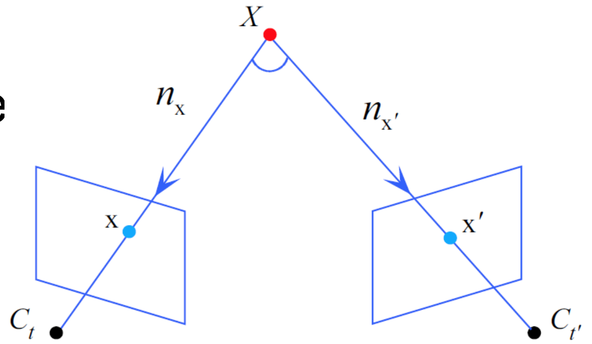
RDSLAM Framework



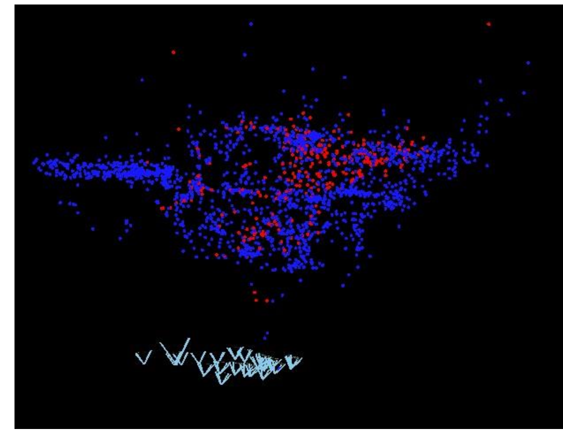
Online 3D Points and Keyframes Updating



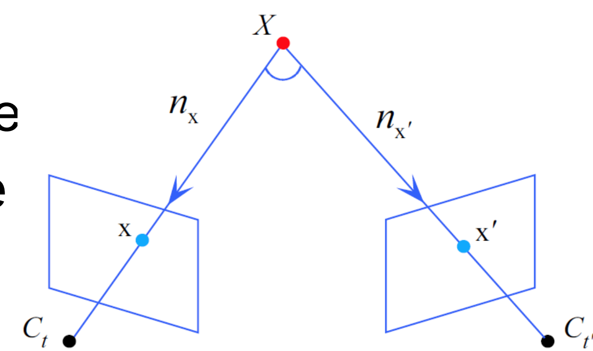
- Keyframe representation
- 3D Change detection
  - Select 5 closest keyframes for online image.
  - For each valid feature point x in each selected keyframe,
    - Compute its projection x' in current frame
    - If $n_{\mathbf{x}}^{\top} \hat{n}_{\mathbf{x}'} < \tau_n$, compute the appearance difference $D_c(X) = \min_d \sum_{\mathbf{y} \in W(\mathbf{x})} |I_{\mathbf{y}} - I_{\mathbf{y}'+d}|$



Online 3D Points and Keyframes Updating

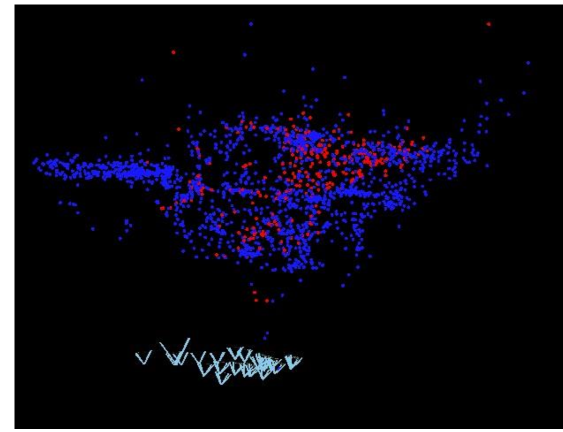


- Keyframe representation
- 3D Change detection
  - Select 5 closest keyframes for online image.
  - For each valid feature point x in each selected keyframe,
    - Compute its projection x’ in current frame
    - If $n_{\mathbf{x}}^{\top} \hat{n}_{\mathbf{x}'} < \tau_n$, compute the appearance difference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



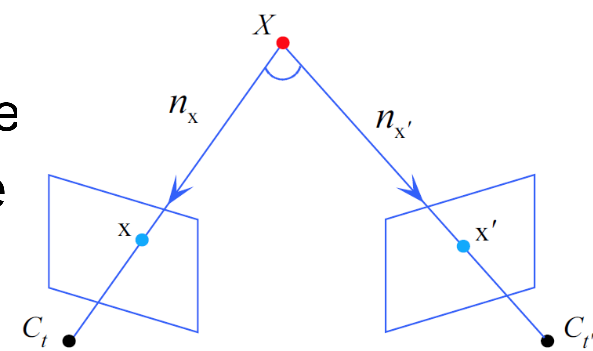
Since dynamic points cannot be triangulated, the occlusion caused by dynamic objects can be excluded here.

Online 3D Points and Keyframes Updating



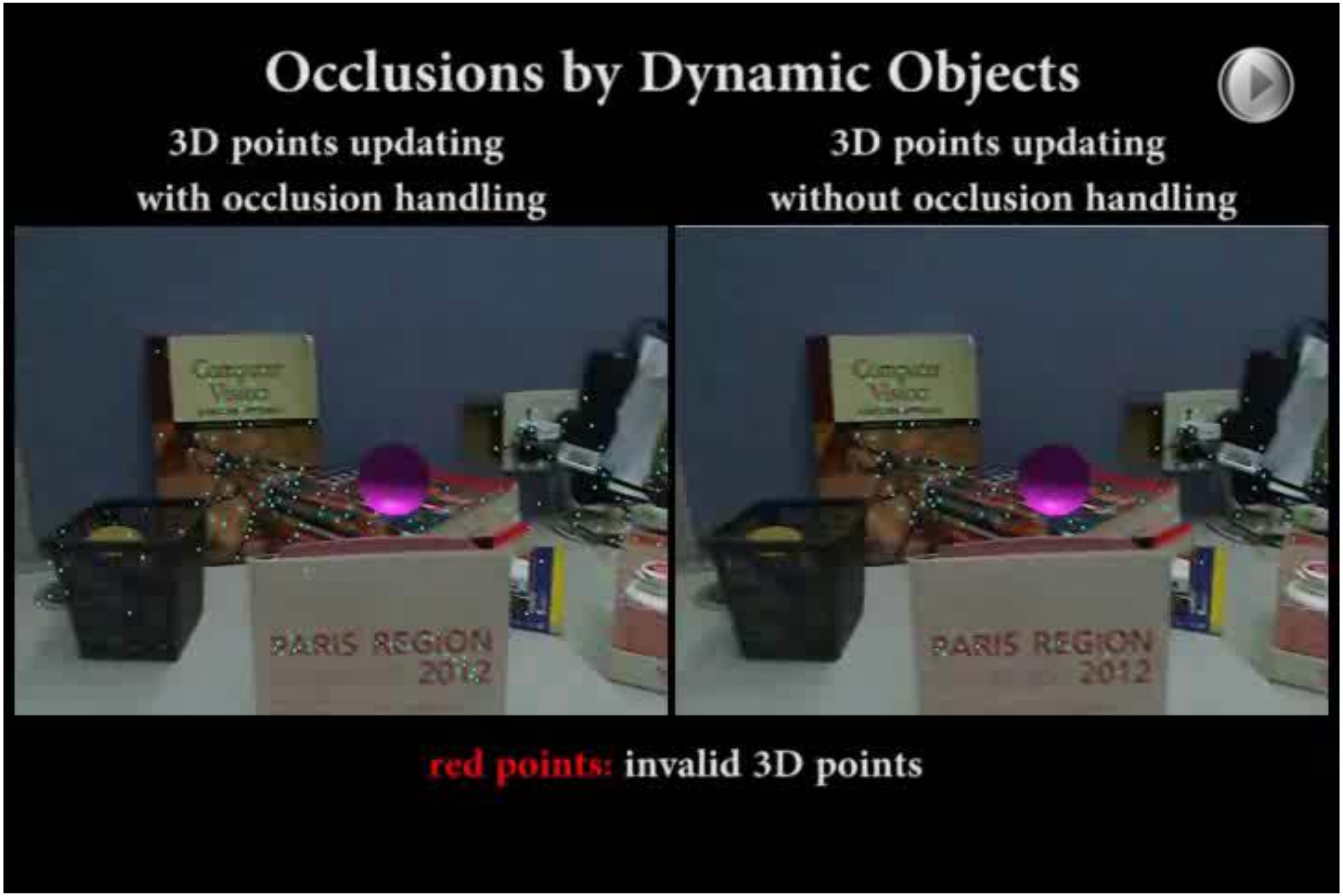
- Keyframe representation
- 3D Change detection
  - Select 5 closest keyframes for online image.
  - For each valid feature point x in each selected keyframe,
    - Compute its projection x' in current frame
    - If $n_{\mathbf{x}}^{\top} \hat{n}_{\mathbf{x}'} < \tau_n$, compute the appearance difference $D_c(X) = \min_{d} \sum_{\mathbf{y} \in W(\mathbf{x})} |I_{\mathbf{y}} - I_{\mathbf{y}'+d}|$
      - 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



Occlusion Handling

(a)
(b)

(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

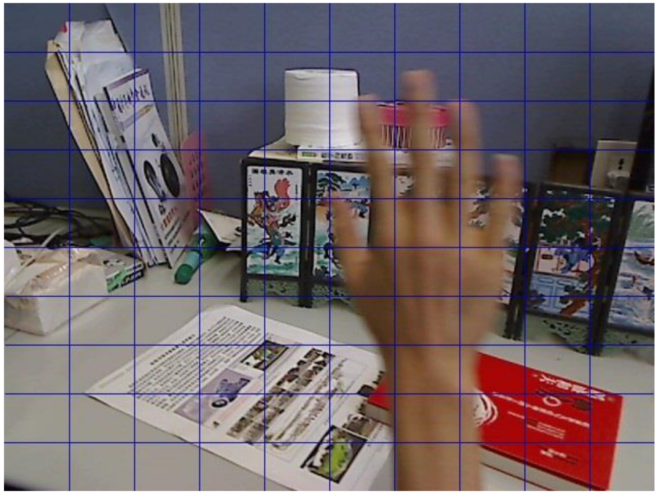
Step 3. Refine *P* based on all inliers

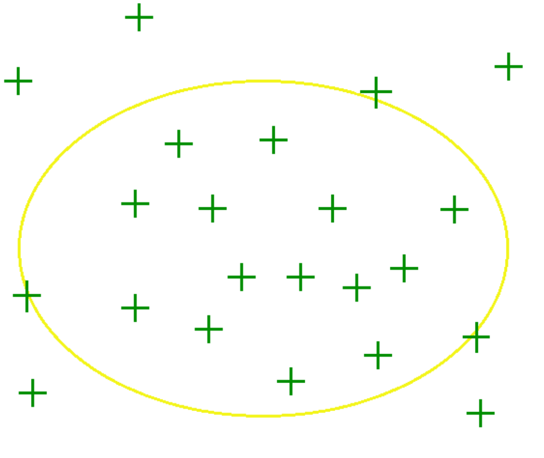
Prior-based Adaptive RANSAC

- 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



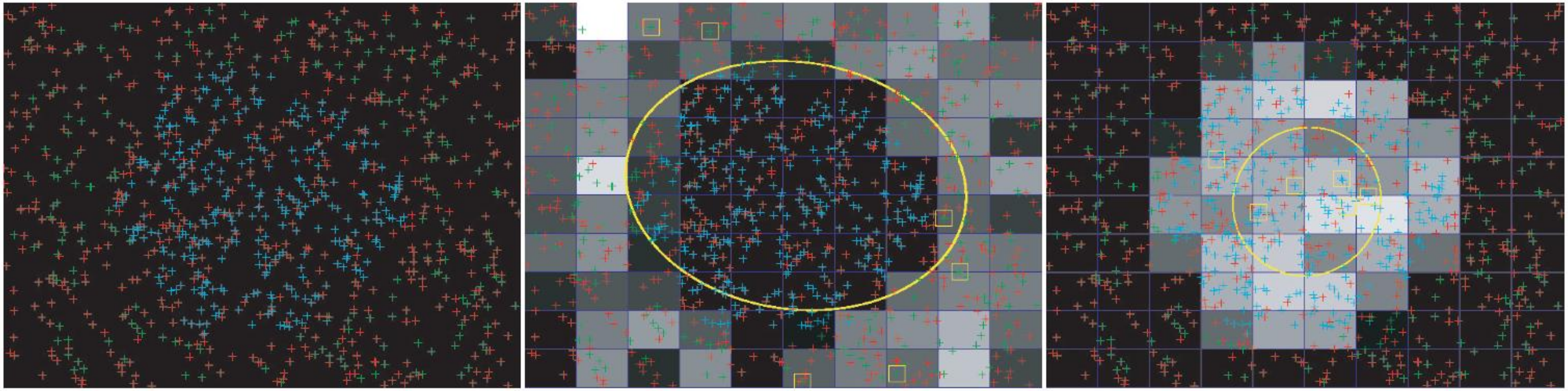


Prior-based Adaptive RANSAC

- Hypothesis evaluation

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

$\sum_i \varepsilon_i$=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.

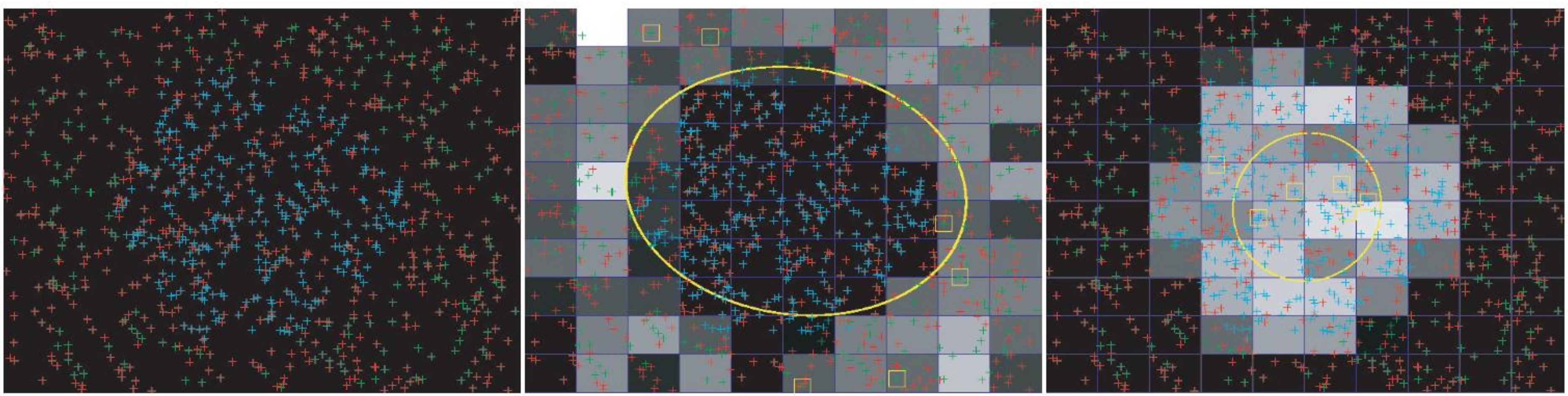
Prior-based Adaptive RANSAC

- Hypothesis evaluation

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

S1 = 8.31 > S2 = 1.98

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

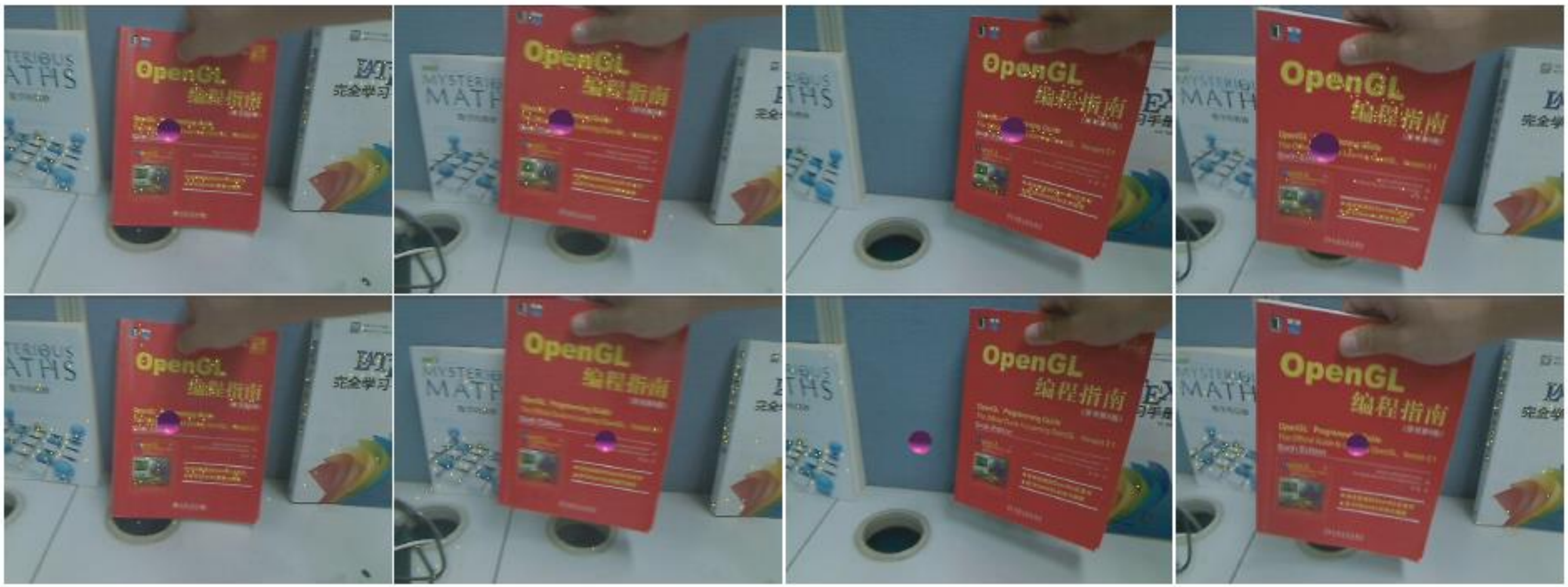


200 green points on the static background, 300 cyan points on the rigidly moving object, 500 red points are randomly moving.

Result Comparison

(a)

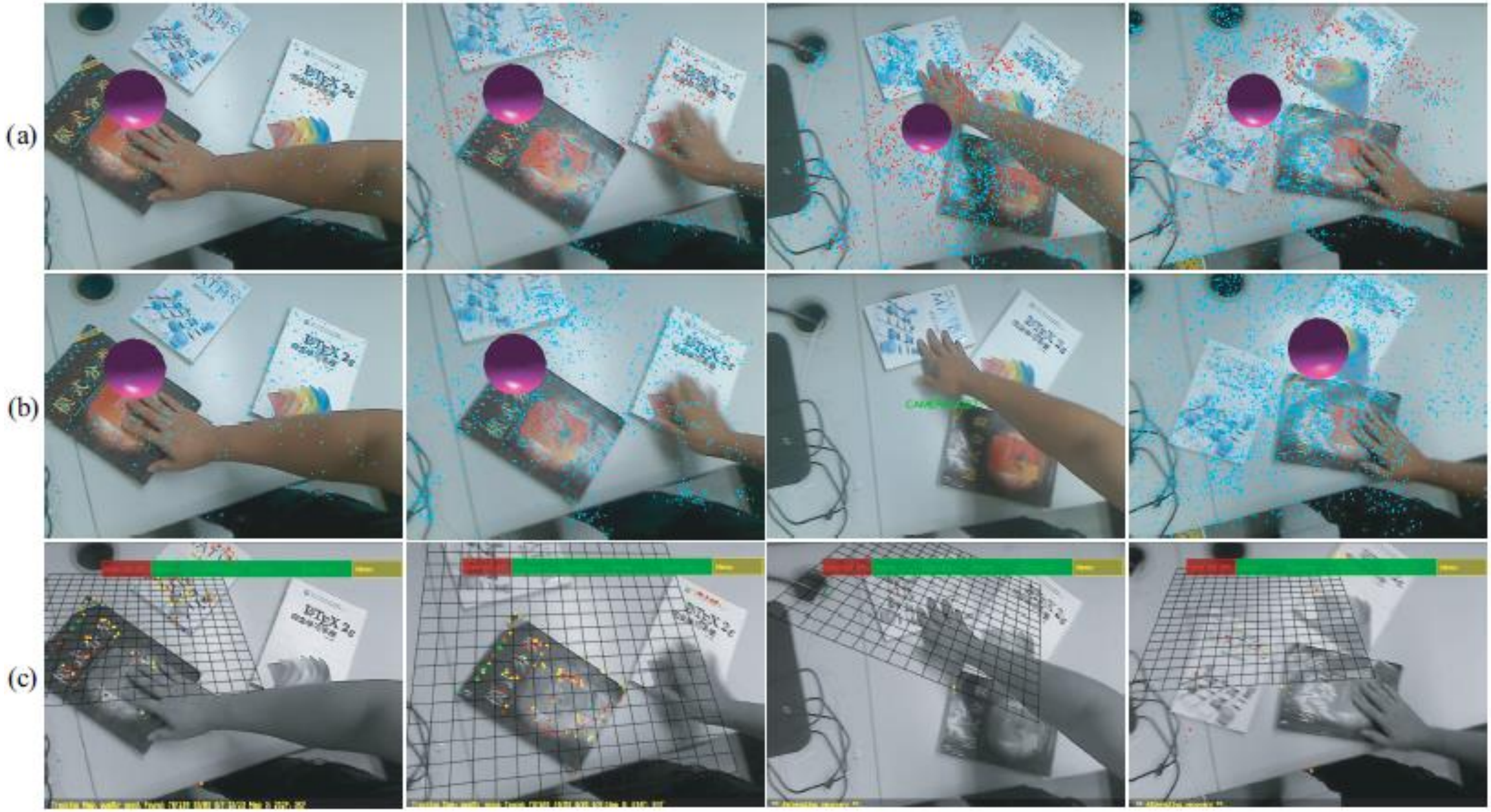
(b)



(a) The SLAM result with standard RANSAC
(b) The SLAM result with our PARSAC

Results and Comparison

Results and Comparison



RDSLAM: Robust Dynamic SLAM

Home

Description

RDSLAM is a real-time simultaneous localization and mapping system which can robustly work in dynamic environments. **It is for non-commercial research and educational use ONLY. Not for reproduction, distribution or commercial use.** If you use this executable for your academic publication, please acknowledge our work. This program is tested on Win7, but is still not guaranteed to be bug-free and work properly with all versions of Windows. You are welcome to report any suggestions or bugs. We will actively update the program. Please email Guofeng Zhang if you have any questions.

Release (RDSLAM1.0 released on Dec. 11, 2013)

RDSLAM1.0 is implemented based on the following paper:

Wei Tan, Haomin Liu, Zilong Dong, Guofeng Zhang* and Hujun Bao. Robust Monocular SLAM in Dynamic Environments. International Symposium on Mixed and Augmented Reality (ISMAR), 2013.

Changelog

http://www.zjucvg.net/rdslam/rdslam.html

Visual-Inertial SLAM

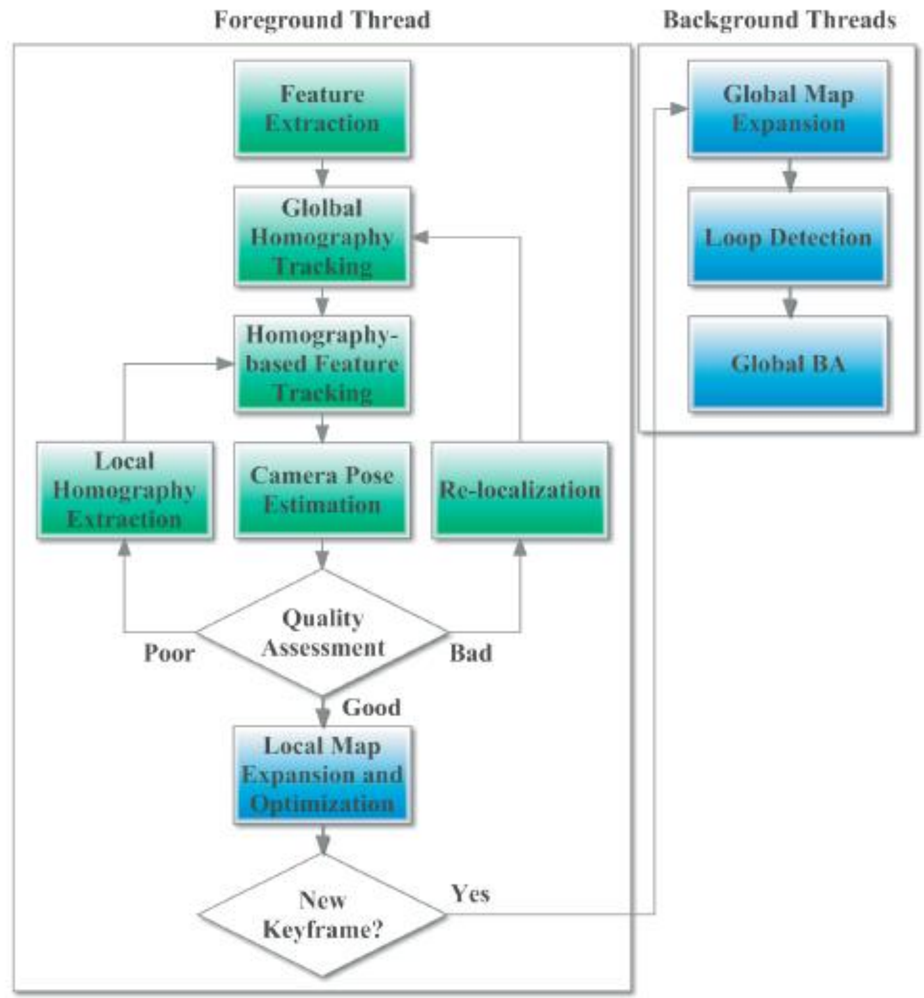
- Use IMU data to improve robustness
  - Filtering-based methods
    - MSCKF, SLAM in Project Tango, …
  - Non-linear optimization based methods
    - OKVIS, …
- Can work without real IMU data?

RKSLAM Framework

- Multi-Homography based Tracking
  - Global homography
  - Specific Homography
  - Local Homographies
- Sliding-window based pose optimization
  - Use global image alignment to estimate rotational velocity

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

  - Pose optimization with simulated IMU data



Multi-Homography based Tracking

- Global Homography Estimation
  - Combine the alignment between the keyframe and previous frame, and the transformation between current frame and previous frame

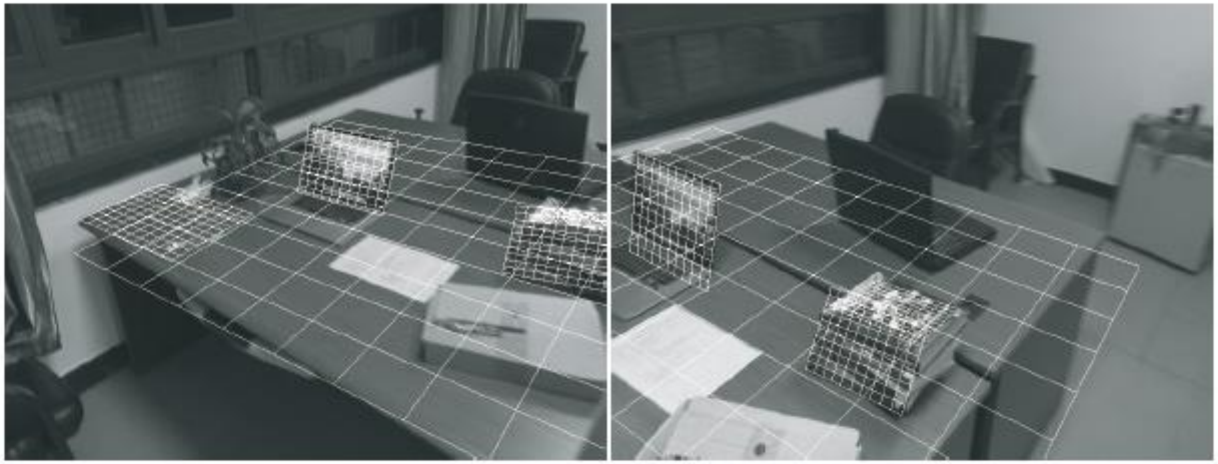
$$\mathbf{H}^G_{k\to(i-1)} = \arg\min_{\mathbf{H}} \Big(\sum_{\mathbf{x}\in\Omega} \|\tilde{F}_k(\mathbf{x}) - \tilde{I}_{i-1}(\pi(\tilde{\mathbf{H}}\mathbf{x}^h))\|_{\delta_I} + \sum_{(\mathbf{x}_k,\mathbf{x}_{i-1})\in M_{k,i-1}} \|\pi(\mathbf{H}\mathbf{x}^h_k) - \mathbf{x}_{i-1}\|_{\delta_\mathbf{x}} \Big).$$

$$\mathbf{H}^G_{k\to i} = \mathbf{H}^G_{(i-1)\to i}\mathbf{H}^G_{k\to(i-1)}$$

Multi-Homography based Tracking

- Specific Homography Estimation
  - For a 3D plane $\mathbf{P}_j$ visible in keyframe F_k, its homography from F_k to I_i can be derived as

$$\mathbf{H}_{k\rightarrow i}^{\mathbf{P}_j} = \mathbf{K}\left(\mathbf{R}_i\mathbf{R}_k^\top + \frac{\mathbf{R}_i(\mathbf{p}_i - \mathbf{p}_k)\mathbf{n}_j^\top\mathbf{R}_k^\top}{d_j + \mathbf{n}_j^\top\mathbf{R}_k\mathbf{p}_k}\right)\mathbf{K}^{-1}$$



Multi-Homography based Tracking

- Local Homography Estimation
  - Same with ENFT algorithm
  - Use the inlier matches to estimate a set of local homographies
- Matching with Multi-Homography
  - Provide better initial positions
  - Alleviate patch distortion
  - Robust to fast motion

Sliding-Window based Pose Optimization

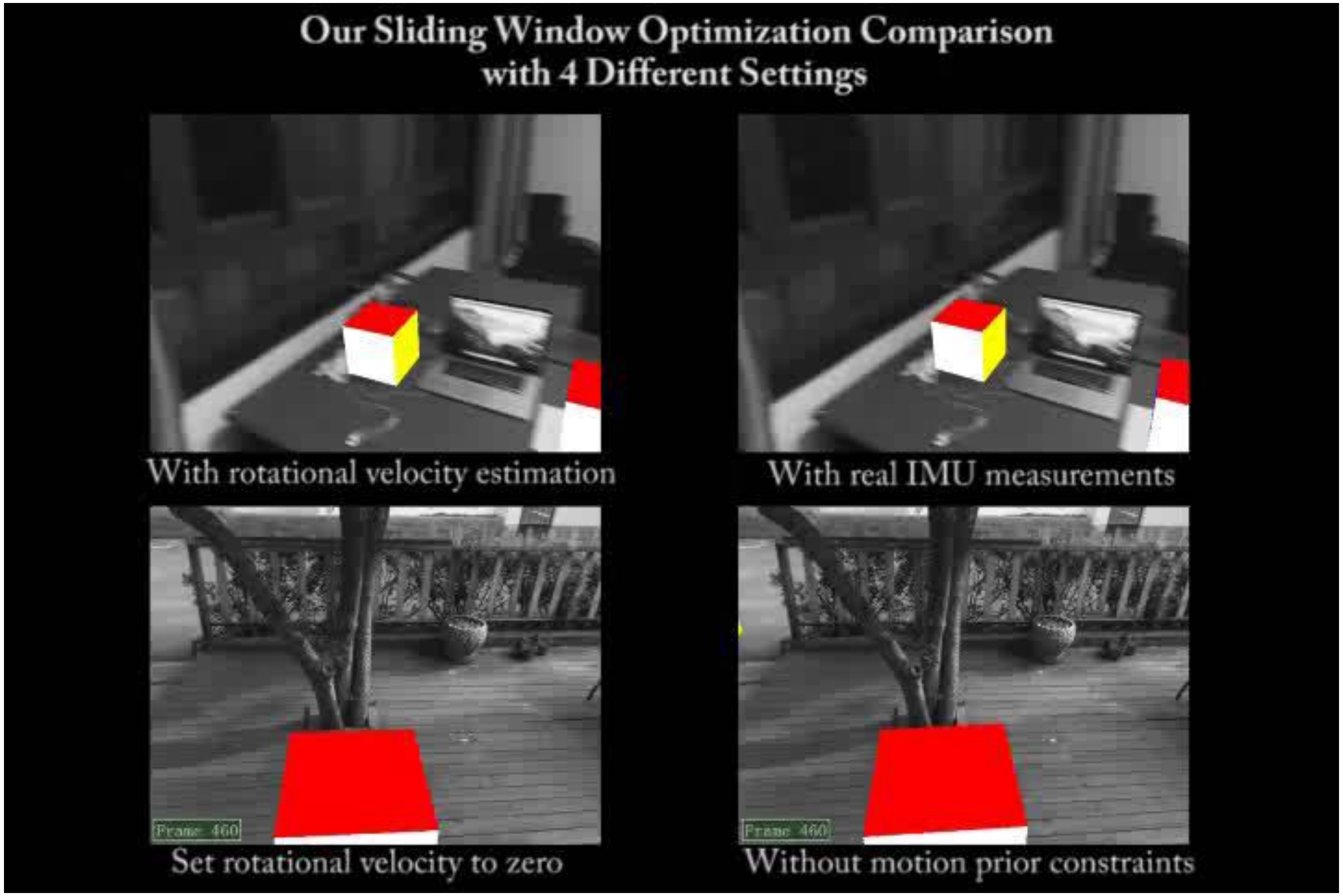
- Assume having IMU data

$$\arg\min_{\mathbf{s}_1\cdots\mathbf{s}_l} \sum_{i=1}^{l} \sum_{j\in V_i} ||\pi(\mathbf{K}(\mathbf{R}_i(\mathbf{X}_j - \mathbf{p}_i))) - \mathbf{x}_{ij}||_{\delta_x} + \sum_{i=1}^{l-1} ||\mathbf{e}_{\mathbf{q}}(\mathbf{q}_i, \mathbf{q}_{i+1}, \mathbf{b}_{\omega_i})||^2_{\Sigma_{\mathbf{q}}}$$
$$+ \sum_{i=1}^{l-1} ||\mathbf{e}_{\mathbf{p}}(\mathbf{q}_i, \mathbf{p}_i, \mathbf{p}_{i+1}, \mathbf{v}_i, \mathbf{b}_{\mathbf{a}_i})||^2_{\Sigma_{\mathbf{p}}} + \sum_{i=1}^{l-1} ||\mathbf{e}_{\mathbf{v}}(\mathbf{q}_i, \mathbf{v}_i, \mathbf{v}_{i+1}, \mathbf{b}_{\mathbf{a}_i})||^2_{\Sigma_{\mathbf{v}}}$$
$$+ \sum_{i=1}^{l-1} ||\mathbf{e}_{\mathbf{b}_\mathbf{a}}(\mathbf{b}_{\mathbf{a}_i}, \mathbf{b}_{\mathbf{a}_{i+1}})||^2_{\Sigma_{\mathbf{b}_\mathbf{a}}} + \sum_{i=1}^{l-1} ||\mathbf{e}_{\mathbf{b}_\omega}(\mathbf{b}_{\omega_i}, \mathbf{b}_{\omega_{i+1}})||^2_{\Sigma_{\mathbf{b}_\omega}}$$

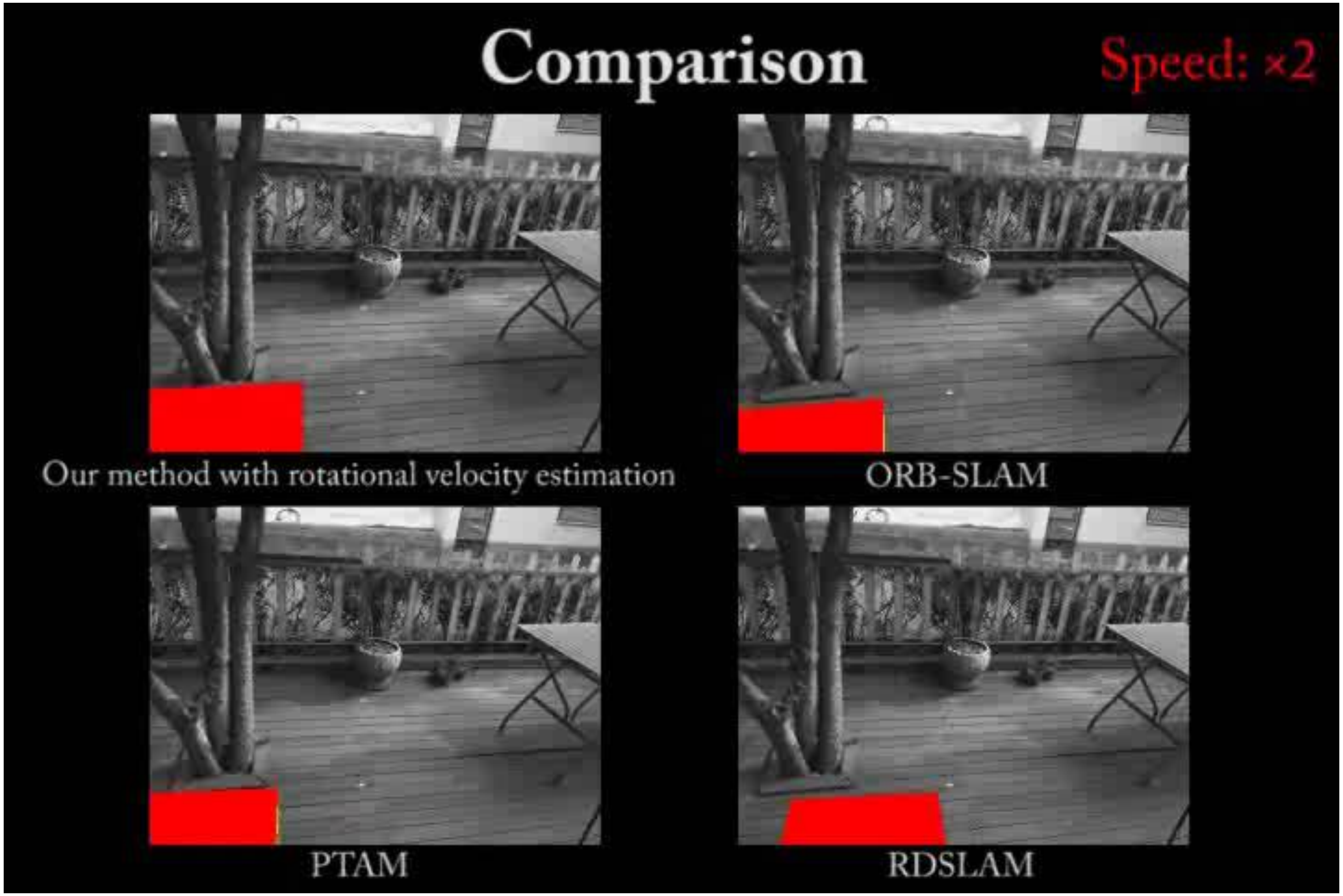
- Set $\hat{\mathbf{a}}_i = 0$ and estimate $\hat{\omega}_i$ by

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

Sliding-Window based Optimization Comparison



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	~ 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.

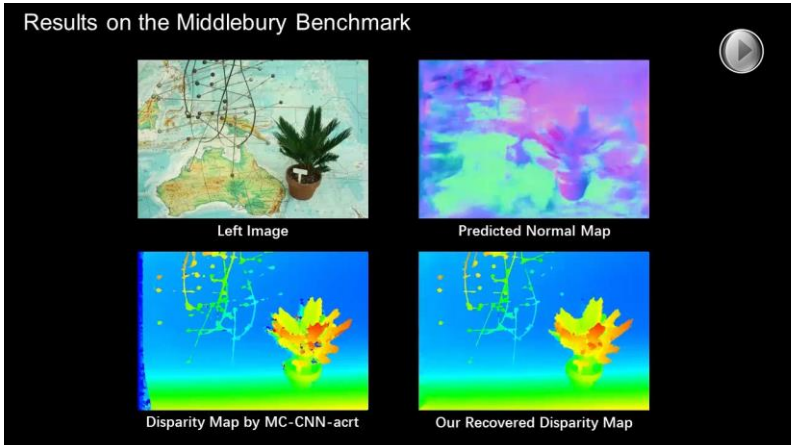
- For a mobile device
  - 20~50 fps on an iPhone 6.

各类单目 V-SLAM 系统比较

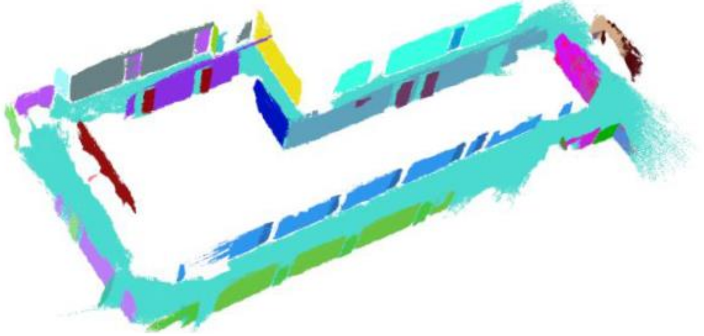
	基于滤波器		基于关键帧 BA			基于直接跟踪	
	MonoSLAM	MSCKF	PTAM	ORB-SLAM	RDSLAM	DTAM	LSD-SLAM
定位精度	✓	✓✓✓	✓✓	✓✓✓	✓✓	✓✓	✓
定位效率	✓	✓✓	✓✓✓	✓✓✓	✓✓	✓✓	✓✓
场景尺度	✓	✓✓✓✓	✓✓	✓✓✓✓	✓✓✓	✓	✓✓✓✓
特征缺失鲁棒性	✓	✓✓✓	✓	✓	✓	✓✓	✓✓
重定位能力	×	×	✓✓	✓✓✓	✓✓✓	✓✓	✓✓✓
快速运动鲁棒性	✓✓	✓✓✓✓	✓✓✓	✓✓✓✓	✓✓✓✓	✓✓✓	✓
扩展效率	✓✓✓	✓✓✓✓	✓✓	✓✓✓	✓✓✓	✓	✓
近似纯旋转扩展鲁棒性	✓✓✓	✓✓✓✓	✓	✓✓	✓	✓	✓
场景变化鲁棒性	✓	✓✓	✓	✓	✓✓✓	✓	✓
回路闭合能力	✓	×	×	✓✓✓	✓✓	×	✓✓✓

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

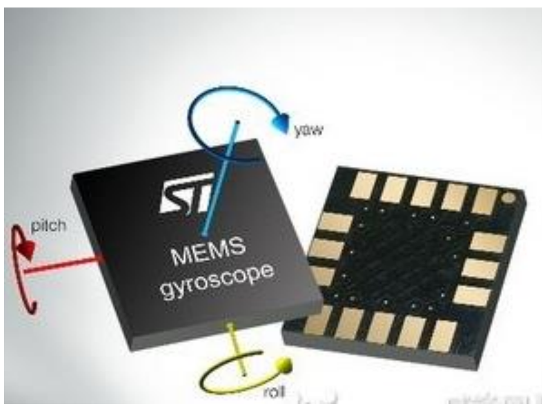


- 稠密三维重建
  - 单／多目实时三维重建
  - 基于深度相机的实时三维重建
  - 平面表达和模型自适应简化



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



我们的SLAM系统

- RDSLAM
  - http://www.zjucvg.net/rdslam/rdslam.html
- RKSLAM
  - http://www.zjucvg.net/rkslam/rkslam.html
- 更多系统未来会放出来
  - http://www.zjucvg.net

推荐开源系统

- PTAM
  - https://github.com/Oxford-PTAM/PTAM-GPL
- ORB-SLAM
  - https://github.com/raulmur/ORB_SLAM
- LSD-SLAM
  - https://github.com/tum-vision/lsd_slam
- DSO
  - https://github.com/JakobEngel/dso
- SVO
  - https://github.com/uzh-rpg/rpg_svo

Thank you!