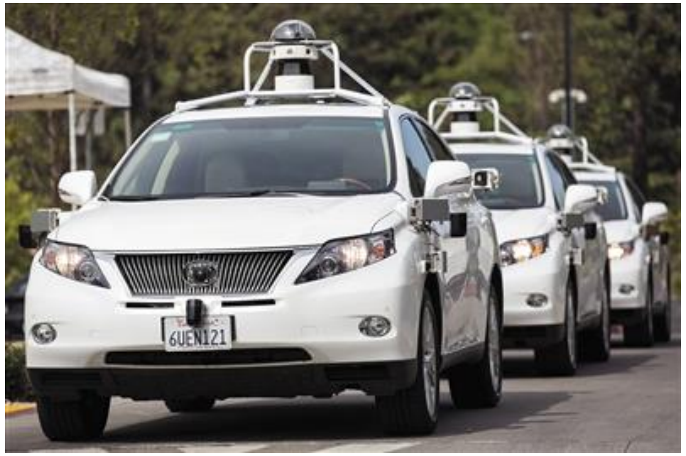
Robust VI-SLAM and HD-Map Reconstruction for Location-based Augmented Reality

Guofeng Zhang

State Key Lab of CAD&CG, Zhejiang University

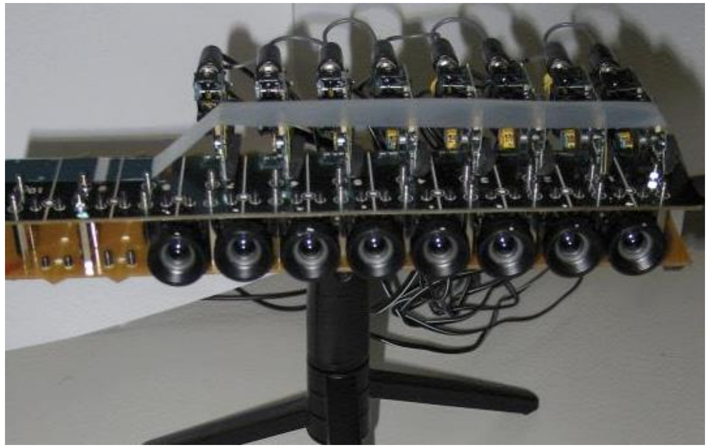
SLAM

- **A Basic Problem in Robotics & Computer Vision**
  - Simultaneously estimate the device pose and 3D scene structure in an unknown environment.
- **Wide Applications**
  - Augmented Reality, Virtual Reality
  - Robotics, Automated Driving

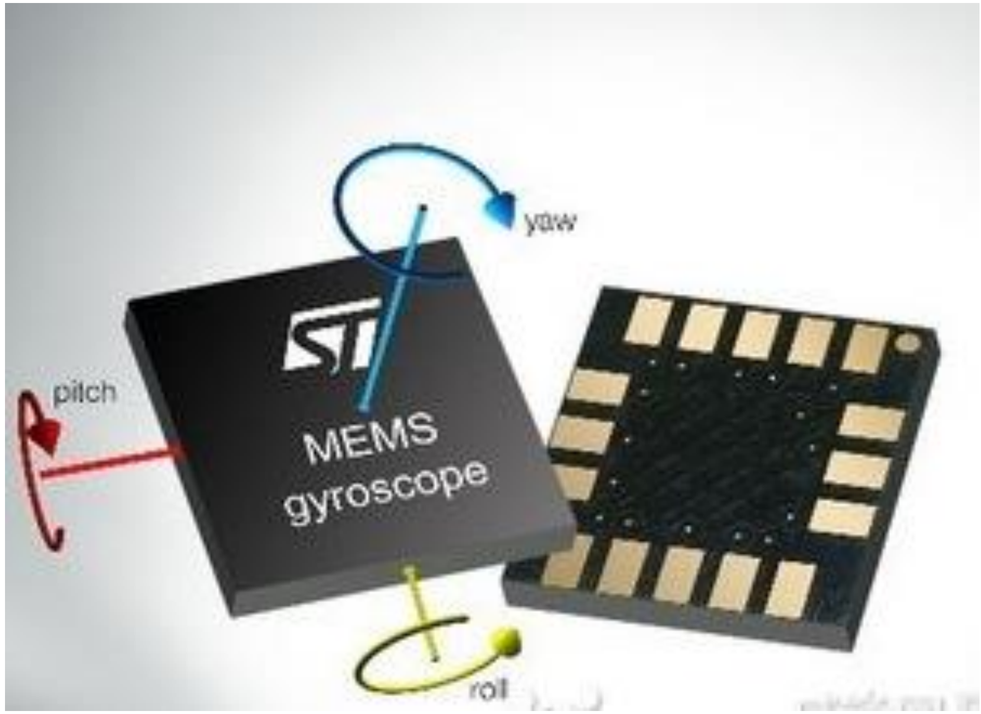


Visual-Inertial SLAM

- **Main Sensors**
  - Single or Multiple Cameras
  - Inertial Measurement Unit



- **Advantages**
  - Low cost
  - High localization accuracy
    at least in a small workspace
  - Inside-out solution: no scene setup

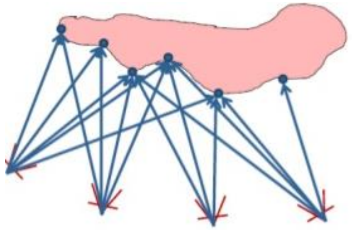


Traditional SLAM Framework

RGB Image Depth Image IMU data

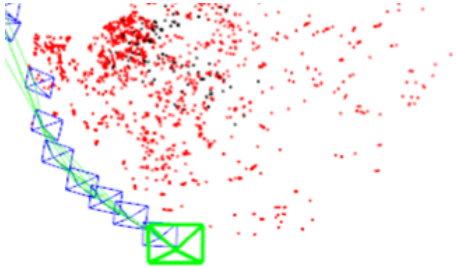
Input

- Sensor data



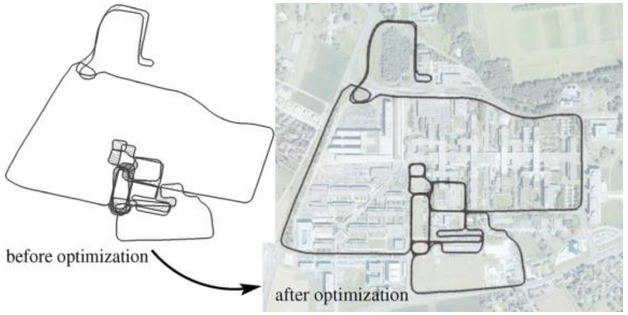
Foreground Thread

- Compute the device pose in real-time



Output

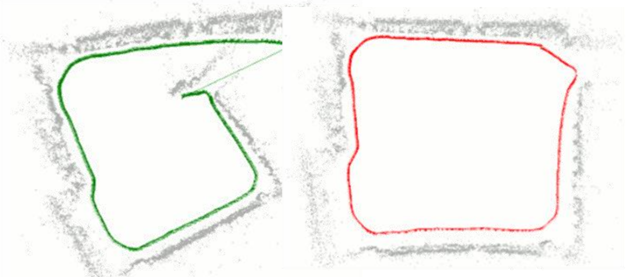
- Device poses
- 3D points



Optimize & reduce accumulation error

Background Thread

- Optimizing local or global maps
- Loop closure detection



Loop closure detection

Challenges of Visual SLAM and VI-SLAM

Challenge 1

Accuracy and Robustness

- Dynamic environment
- Lots of outliers
- Unstable optimization

Challenge 2

Real-Time Performance

- Large scene
- Huge computation
- Limited computation resources on a mobile device

Key Idea for Robust Estimation

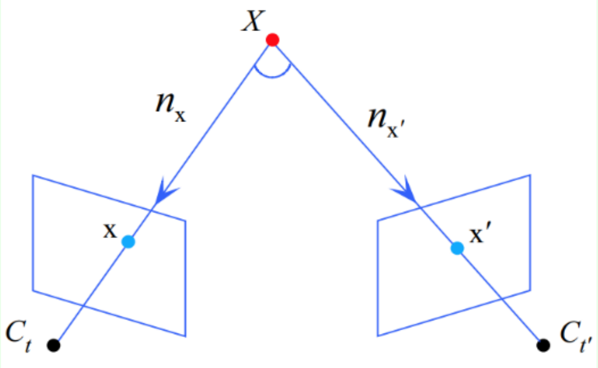
Robust Estimation

Accuracy of Constraints

Sufficiency of Constraints

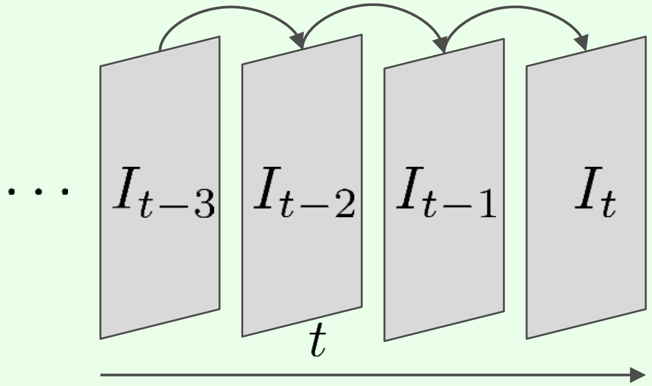
outliers detection and removement

distribution prior

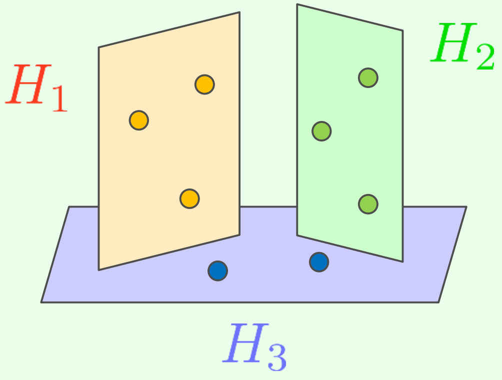


change detection

motion priors



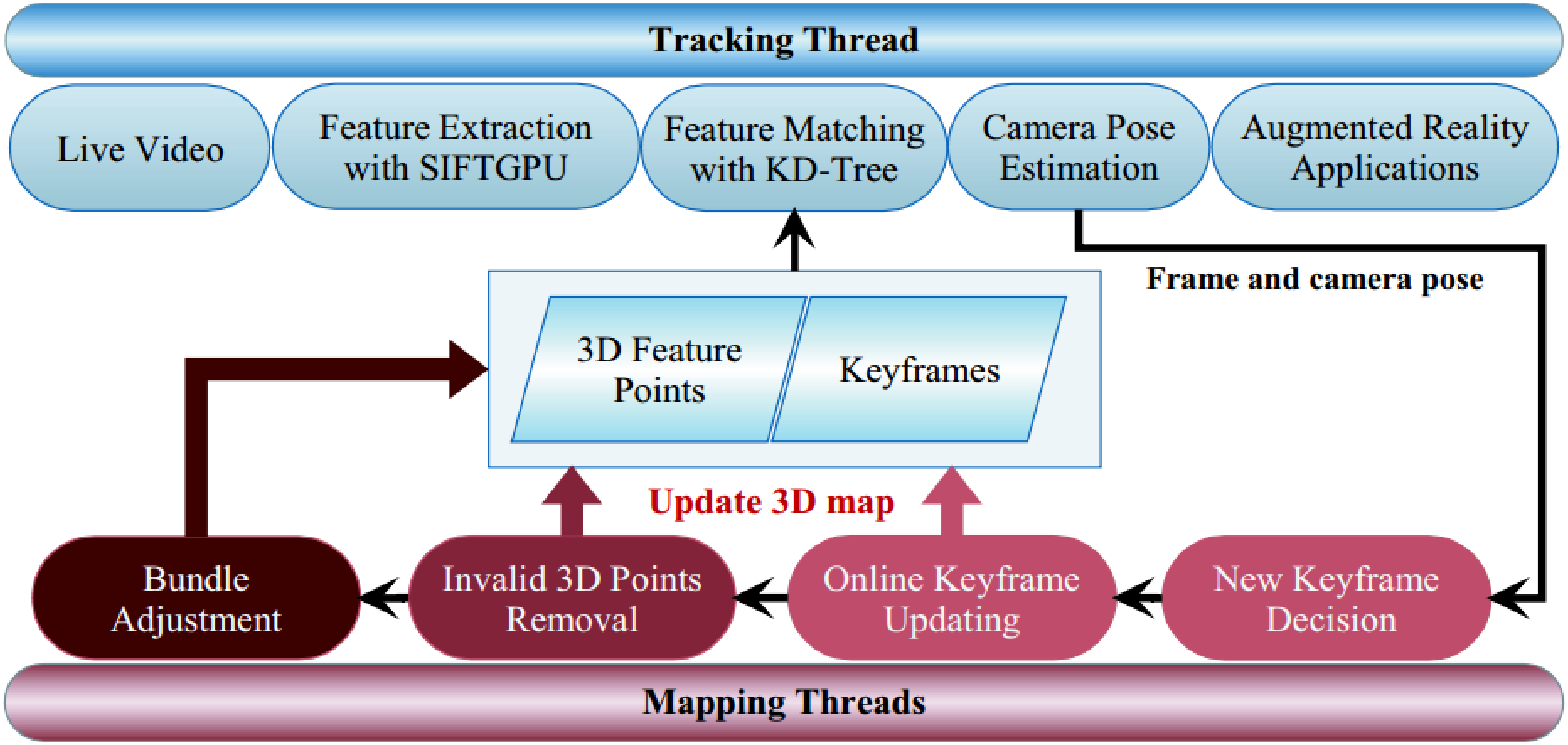
structure priors



Outliers Detection and Removement

ISMAR 2013

- Detect the changed 3D points and update the keyframes adaptively



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.

RDSLAM

Visual-Inertial Odometry with Multi-Plane Priors

PRCV 2019

- Motivation
  - A fast and lightweight VIO method for low-end mobile devices.
  - Planes commonly exist in human-made scenes, and can be utilized.
- Plane extraction and expansion from VIO point cloud
  - Reprojection Consensus
- Plane Constraints
  - VIP-PnP : solving the BA as if some points lying on planes
  - Structureless Plane-Distance Error

$$\epsilon_k = \sum_i \|u_{ik}(\lambda_k) - \tilde{u}_{ik}\|^2, \quad \epsilon_k^{\perp} = \sum_i \|u_{ik}(\lambda_k^{\perp}) - \tilde{u}_{ik}\|^2$$

Ordinary RPE | Point-on-Plane RPE

$$A_{sk} = \begin{pmatrix} A_k \\ w_k n_s^{\top} \end{pmatrix}, \quad b_{sk} = \begin{pmatrix} b_k \\ w_k d_s \end{pmatrix}$$

Augmented triangulation with Plane Constraint

$$x_{sk} = (A_{sk}^{\top} A_{sk})^{-1} A_{sk}^{\top} b_{sk}$$

$$r_P(\{{}_b^w p_i, {}_b^w q_i\}, n_s, d_s) = |n_s^{\top} x_{sk} - d_s|$$

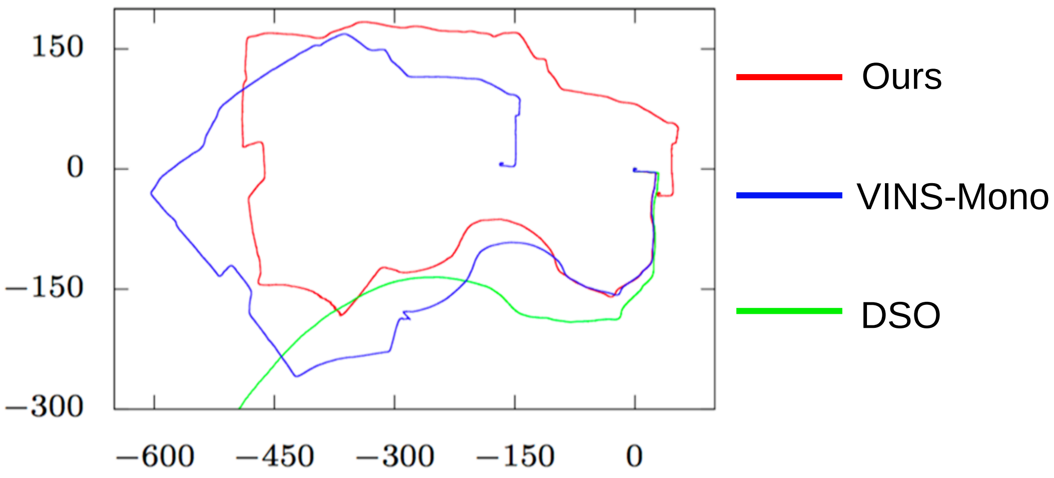
Minimize Point-to-Plane Distance Error

- Augment for degenerated constraints (insufficient parallax/observations)
- For 1 landmark, *m* reprojection error → 1 structureless error

A Robust and Efficient Visual-Inertial Odometry with Multi-Plane Priors. Jinyu Li, Bangbang Yang, Kai Huang, Guofeng Zhang, Hujun Bao*. Chinese Conference on Pattern Recognition and Computer Vision, 2019.*

Experimental Results

Trajectories on TUM-VI Outdoors1



	Dataset	ORB-SLAM2		SVO2		DSO	VINS-Mono		PVIO	
		−Loop	+Loop	E+P	BA		−Loop	+Loop	−Plane	+Plane
EuRoC [2]	MH_01	0.02	0.03	0.10	0.06	0.05	0.16	0.15	0.19	**0.13**
	MH_02	0.03	0.03	0.12	0.07	0.05	0.18	0.26	**0.16**	0.21
	MH_03	0.17	0.05	0.41	×	0.18	0.20	**0.11**	0.31	0.16
	MH_04	0.15	0.37	0.43	0.40	2.50	0.35	0.37	0.29	**0.29**
	MH_05	0.06	0.04	0.30	×	0.11	0.30	**0.28**	0.79	0.34
	V1_01	0.03	0.03	0.07	0.05	0.12	0.09	0.10	0.10	**0.08**
	V1_02	0.15	0.03	0.21	×	0.11	0.11	0.09	×	**0.09**
	V1_03	(0.49)	0.10	×	×	0.93	0.19	0.18	×	**0.16**
	V2_01	0.03	0.03	0.11	×	0.04	0.09	0.08	0.11	**0.05**
	V2_02	0.15	0.03	0.11	×	0.13	**0.16**	0.17	×	0.20
	V2_03	(0.73)	(0.40)	1.08	×	1.16	0.29	0.37	×	**0.29**
TUM-VI [21]	Room1	×	0.10	×	×	0.06	0.07	**0.07**	1.65	0.26
	Room2	×	0.12	×	×	0.11	0.07	**0.07**	0.12	0.15
	Room3	×	(0.04)	×	×	0.12	0.12	**0.12**	0.18	0.18
	Corridor1	×	×	×	×	5.43	0.59	0.59	×	**0.23**
	Outdoors1	×	×	×	×	×	74.55	81.57	×	**22.26**

Single thread

No loop-closure & relocalization

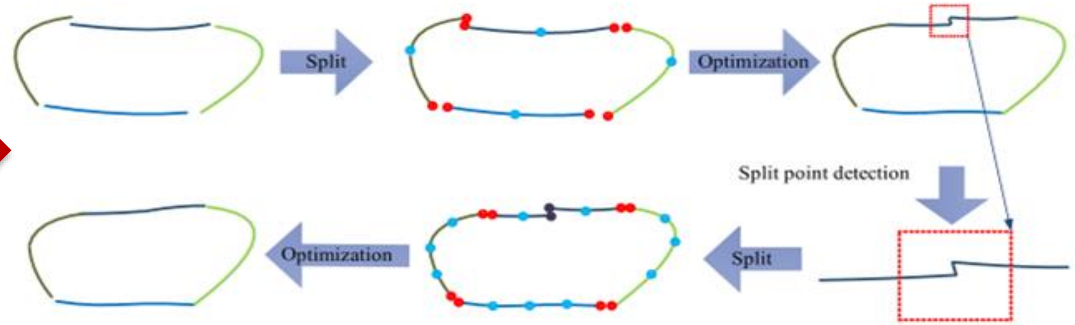
Key Idea for Efficient Estimation

- **Bundle Adjustment**
  - Jointly optimize all cameras and points
  - Time-consuming and require large memory space

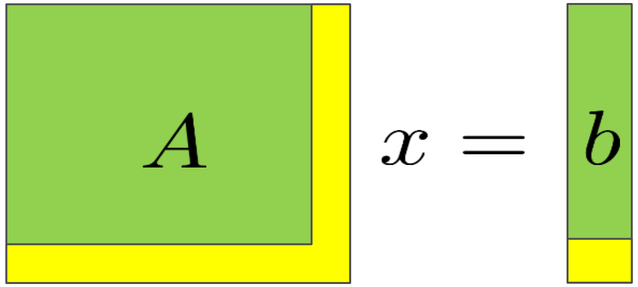
$$\mathop{\arg\min}_{C_1,\ldots C_{N_c}, X_1,\ldots,X_{N_p}} \sum \left\| \pi(X_i, C_j) - x_{ij} \right\|^2$$

Efficient Estimation

Coarse-to-Fine Strategy



Incremental Computation



Sparse Bundle Adjustment

$$\underset{C_1,\dots C_{N_c}, X_1,\dots,X_{N_p}}{\arg\min} \sum \left\| \pi(X_i, C_j) - x_{ij} \right\|^2$$

1 Point

1 Camera

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} d_C \\ d_X \end{pmatrix} = - \begin{pmatrix} u \\ v \end{pmatrix}$$

$$\begin{pmatrix} U - WV^{-1}W^T & 0 \\ W^T & V \end{pmatrix} \begin{pmatrix} d_C \\ d_X \end{pmatrix} = - \begin{pmatrix} u - WV^{-1}v \\ v \end{pmatrix}$$

$$S = U - WV^{-1}W^T$$

$$Sd_C = -(u - WV^{-1}v)$$

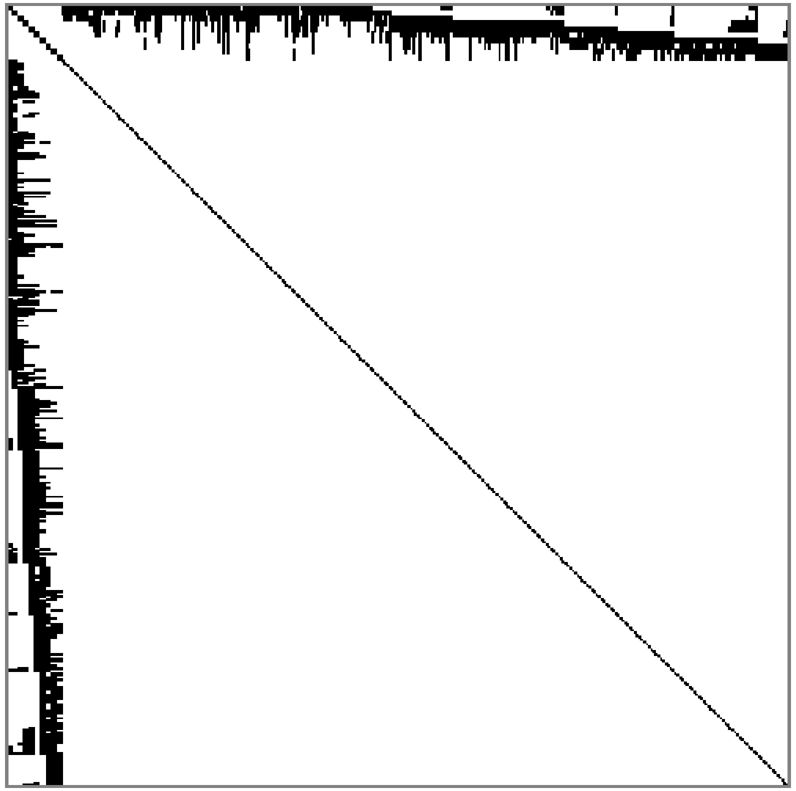
$$Vd_X = -v - W^T d_C$$

Schur Complement

Compute cameras first (# cameras << # points)

back substitution for points

Sparsity patten of Hessian



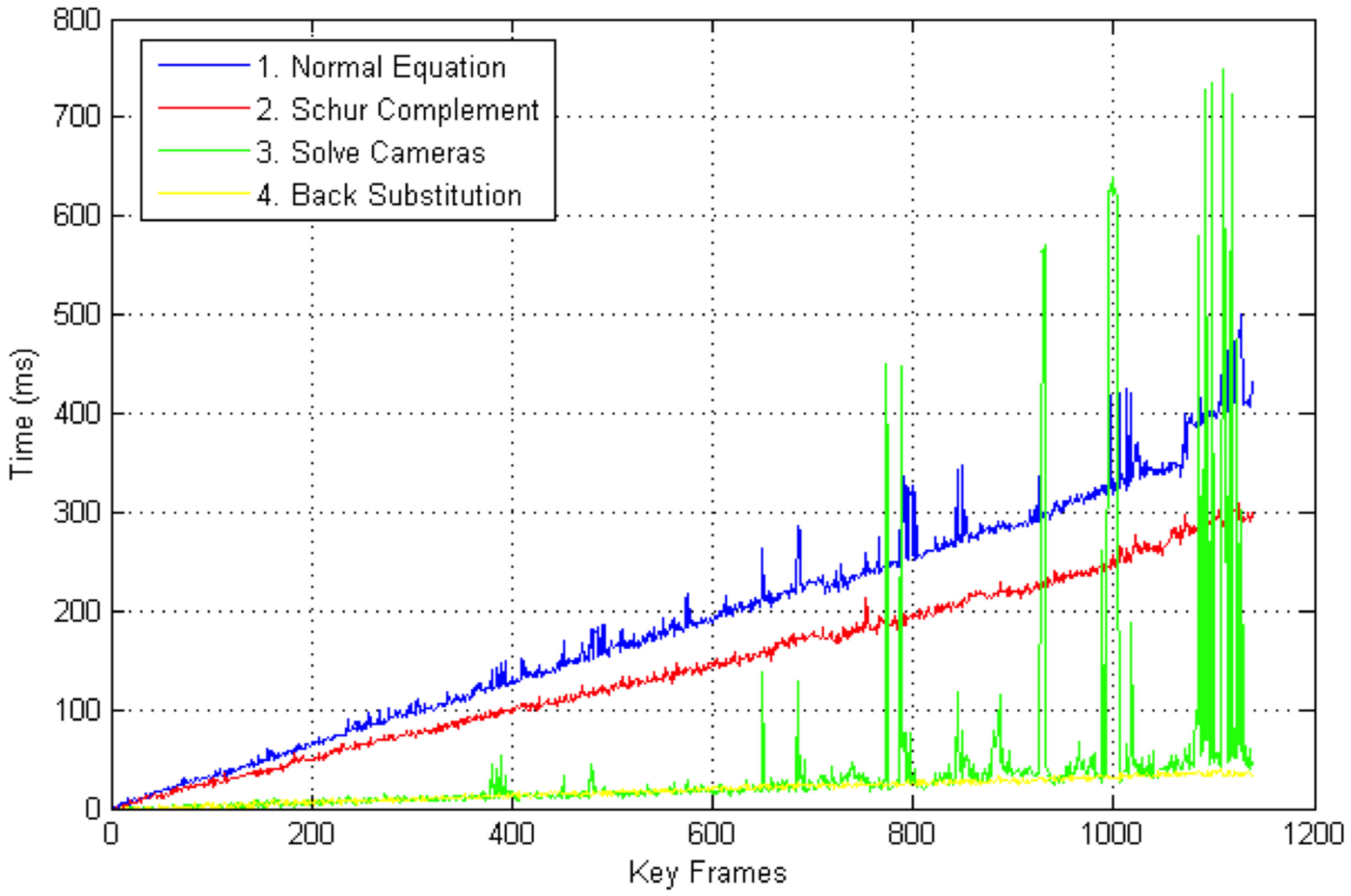
Manolis I. A. Lourakis, Antonis A. Argyros: SBA: A software package for generic sparse bundle adjustment. ACM Trans. Math. Softw. 36(1) (2009)

Sparse Bundle Adjustment

- Runtime significantly increases with the number of cameras

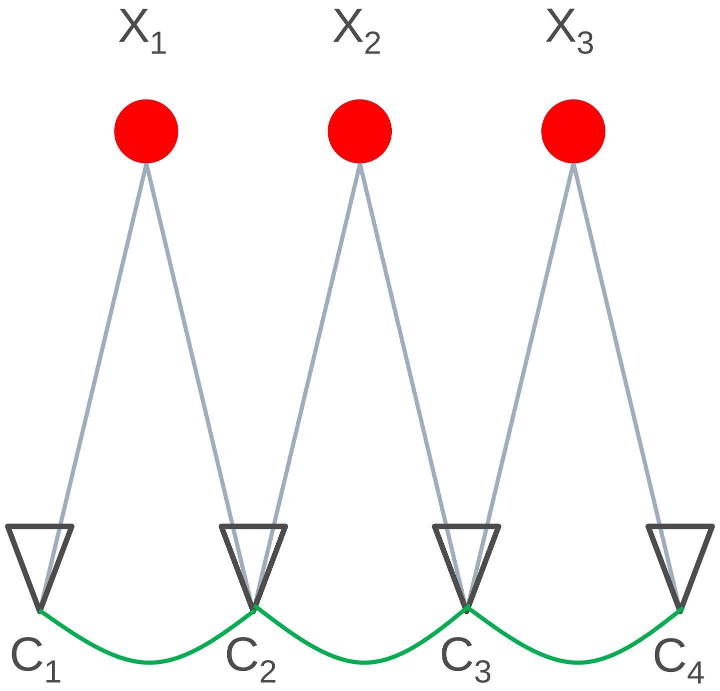
Four Steps in one iteration

1. Normal equation
2. Schur complement
3. Solve cameras
4. Solve points by back substitution



Batch VS Incremental BA

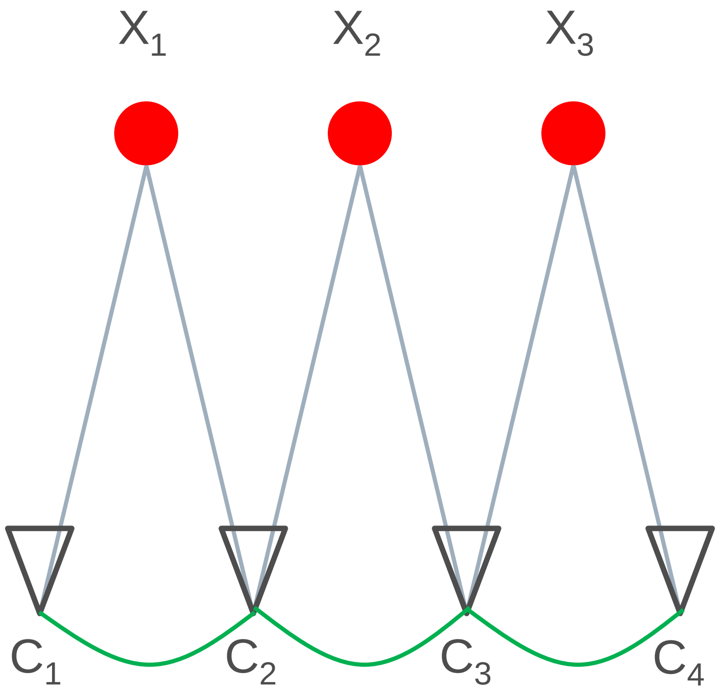
- Batch BA



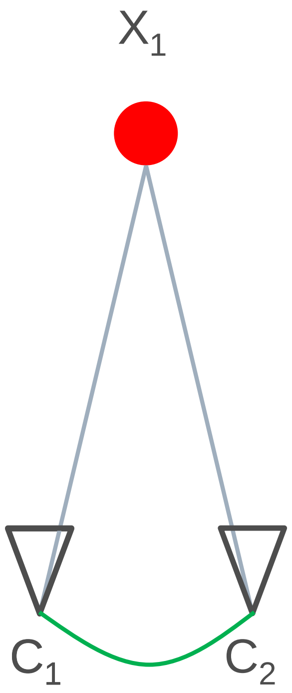
- Incremental BA

Batch VS Incremental BA

- Batch BA

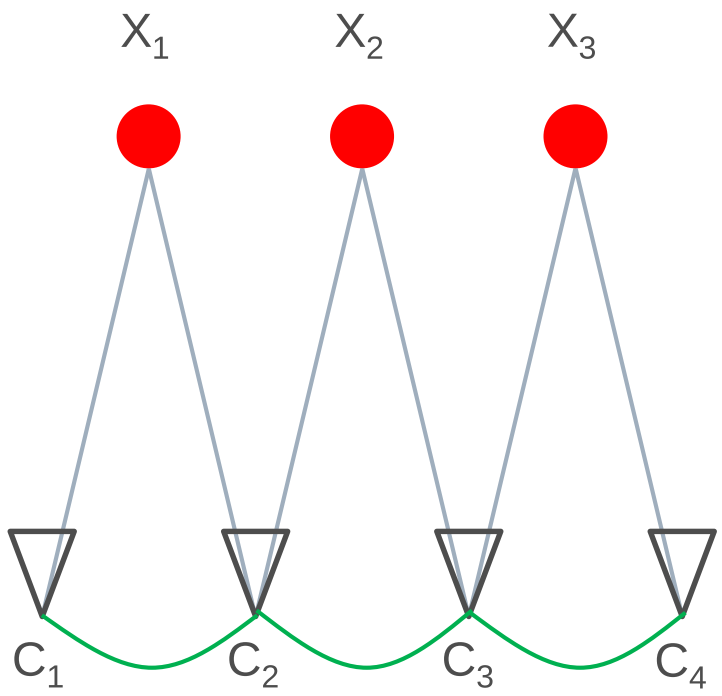


- Incremental BA

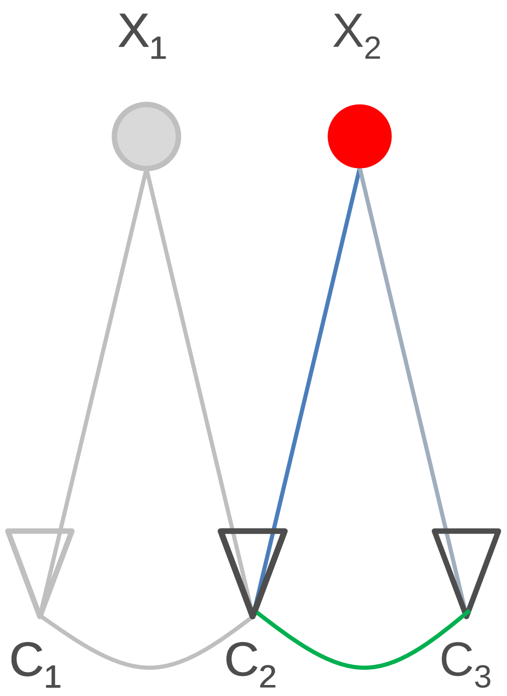


Batch VS Incremental BA

- Batch BA

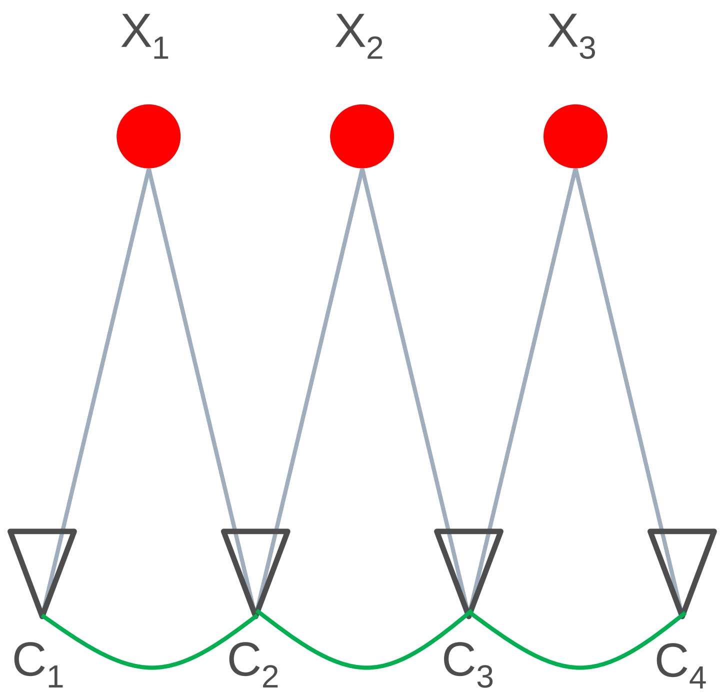


- Incremental BA

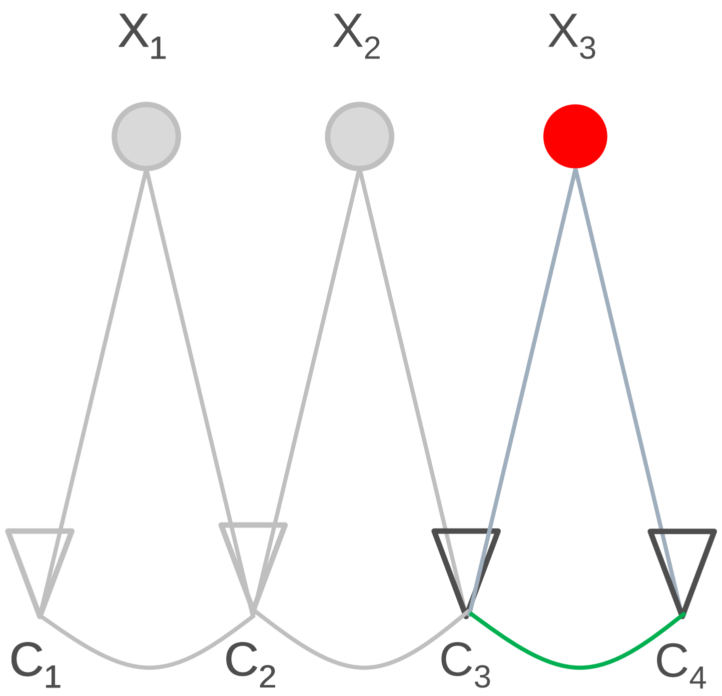


Batch VS Incremental BA

- Batch BA



- Incremental BA



Incremental Bundle Adjustment

- **Most cameras and points are nearly unchanged**
  - Contribution of most projection functions nearly remains the same
  - No need to re-compute at each iteration
- **Incremental approaches**
  - iSAM, iSAM2, SLAM++
  - Our EIBA & ICE-BA
- **Key ideas**
  - Incremental update: makes maximum use of intermediate computation for efficiency
  - Detect the actually changed variables and adaptively update them

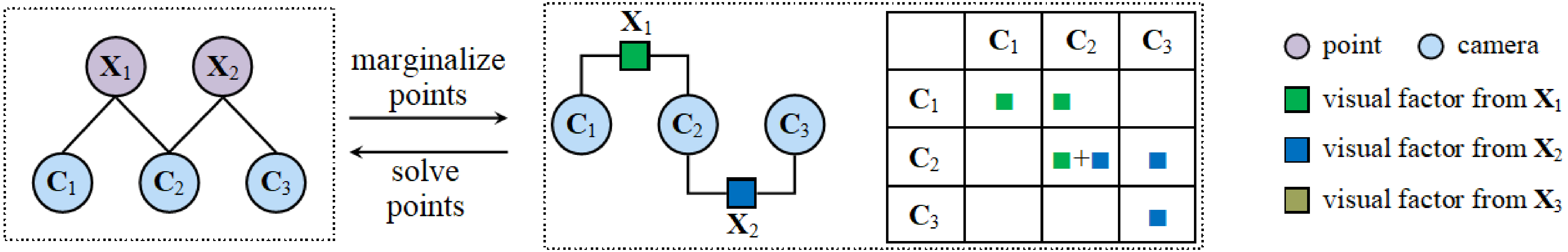
$$[\mathbf{A}|\mathbf{b}] = \left[\begin{array}{cc|c} \mathbf{U} & \mathbf{W} & \mathbf{u} \\ \mathbf{W}^T & \mathbf{V} & \mathbf{v} \end{array}\right]$$

$$[\mathbf{A}|\mathbf{b}]^+ = [\mathbf{A}|\mathbf{b}]^- + \left[\ \textstyle\sum_{k\in\mathcal{L}} \delta\mathbf{A}_k \ \middle|\ \textstyle\sum_{k\in\mathcal{L}} \delta\mathbf{b}_k\ \right]$$

ICE-BA: Incremental, Consistent and Efficient BA for VI-SLAM

CVPR 2018

- Factor graph representation

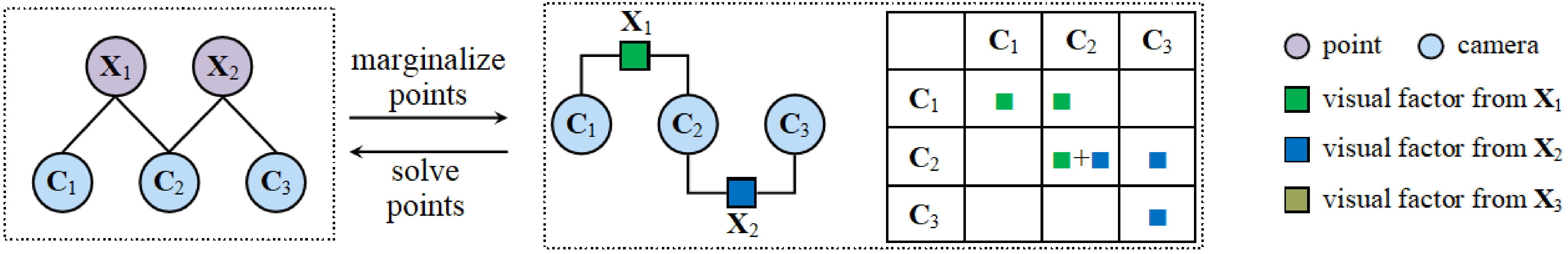


Haomin Liu, Mingyu Chen, Guofeng Zhang, Hujun Bao and Yingze Bao. ICE-BA: Incremental, Consistent and Efficient Bundle Adjustment for Visual-Inertial SLAM. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

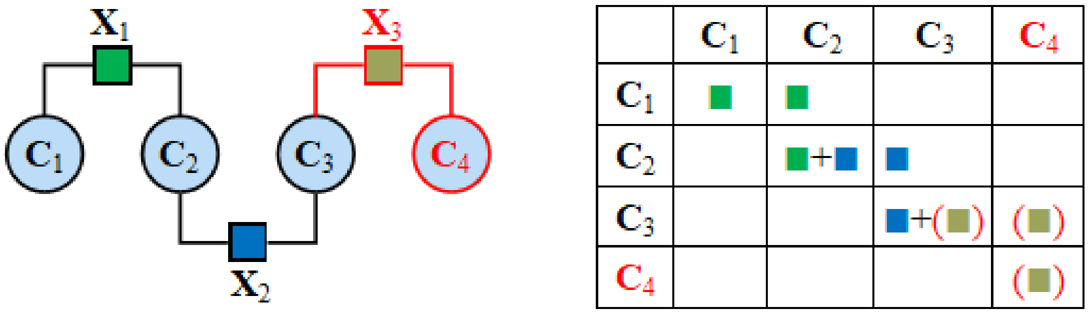
ICE-BA: Incremental, Consistent and Efficient BA for VI-SLAM

CVPR 2018

- Factor graph representation



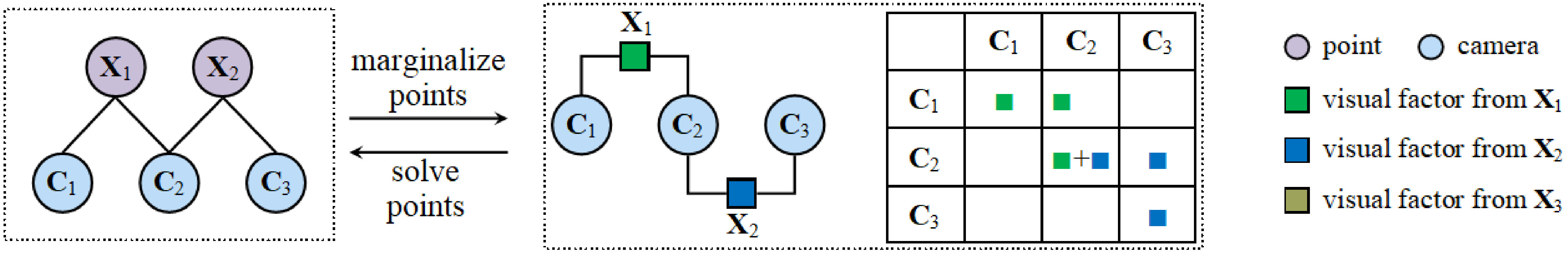
- New cameras or points come



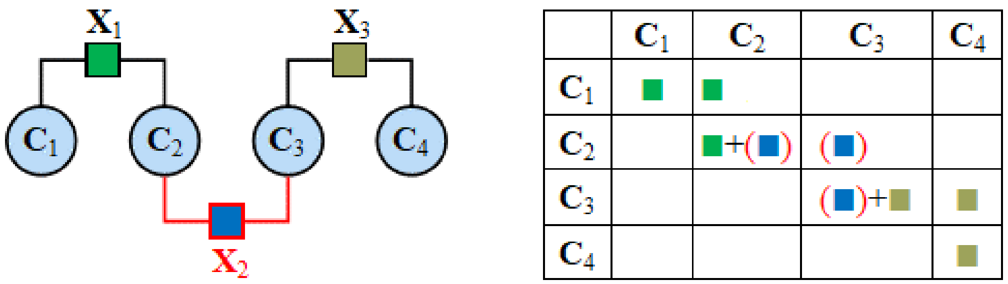
ICE-BA: Incremental, Consistent and Efficient BA for VI-SLAM

CVPR 2018

- Factor graph representation



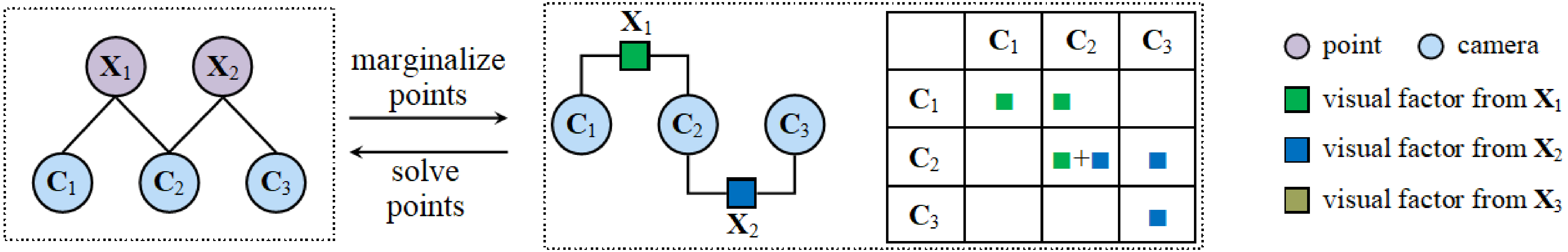
- Points have changed after iteration



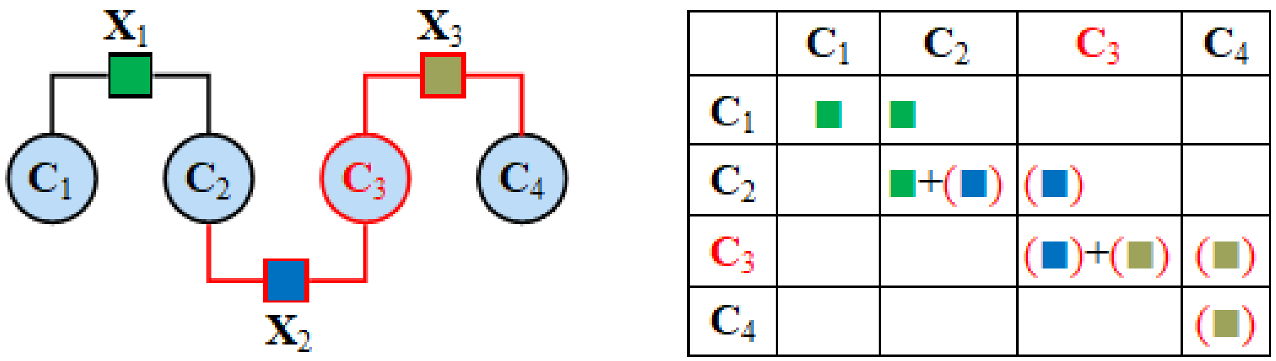
ICE-BA: Incremental, Consistent and Efficient BA for VI-SLAM

CVPR 2018

- Factor graph representation



- Cameras have changed after iteration



Step1: Normal Equation

- Batch BA

$\mathbf{U} = \mathbf{0}$; $\mathbf{V} = \mathbf{0}$; $\mathbf{W} = \mathbf{0}$; $\mathbf{u} = \mathbf{0}$; $\mathbf{v} = \mathbf{0}$
for each point j and each camera $i \in \mathcal{V}_j$ **do**
  Construct linearized equation (11)
  $\mathbf{U}_{ii} += \mathbf{J}_{\mathbf{C}_{ij}}^{\top} \mathbf{J}_{\mathbf{C}_{ij}}$
  $\mathbf{V}_{jj} += \mathbf{J}_{\mathbf{X}_{ij}}^{\top} \mathbf{J}_{\mathbf{X}_{ij}}$
  $\mathbf{u}_{i} += \mathbf{J}_{\mathbf{C}_{ij}}^{\top} \mathbf{e}_{ij}$
  $\mathbf{v}_{j} += \mathbf{J}_{\mathbf{X}_{ij}}^{\top} \mathbf{e}_{ij}$
  $\mathbf{W}_{ij} = \mathbf{J}_{\mathbf{C}_{ij}}^{\top} \mathbf{J}_{\mathbf{X}_{ij}}$
end for

- ICE-BA

for each point j and each camera $i \in \mathcal{V}_j$ that $\mathbf{C}_i$ or $\mathbf{X}_j$ is changed **do**
  Construct linearized equation (11)
  $\mathbf{S}_{ii} -= \mathbf{A}_{ij}^{\mathbf{U}}$; $\mathbf{A}_{ij}^{\mathbf{U}} = \mathbf{J}_{\mathbf{C}_{ij}}^{\top} \mathbf{J}_{\mathbf{C}_{ij}}$; $\mathbf{S}_{ii} += \mathbf{A}_{ij}^{\mathbf{U}}$
  $\mathbf{V}_{jj} -= \mathbf{A}_{ij}^{\mathbf{V}}$; $\mathbf{A}_{ij}^{\mathbf{V}} = \mathbf{J}_{\mathbf{X}_{ij}}^{\top} \mathbf{J}_{\mathbf{X}_{ij}}$; $\mathbf{V}_{jj} += \mathbf{A}_{ij}^{\mathbf{V}}$
  $\mathbf{g}_{i} -= \mathbf{b}_{ij}^{\mathbf{u}}$; $\mathbf{b}_{ij}^{\mathbf{u}} = \mathbf{J}_{\mathbf{C}_{ij}}^{\top} \mathbf{e}_{ij}$; $\mathbf{g}_{i} += \mathbf{b}_{ij}^{\mathbf{u}}$
  $\mathbf{v}_{j} -= \mathbf{b}_{ij}^{\mathbf{v}}$; $\mathbf{b}_{ij}^{\mathbf{v}} = \mathbf{J}_{\mathbf{X}_{ij}}^{\top} \mathbf{e}_{ij}$; $\mathbf{v}_{j} += \mathbf{b}_{ij}^{\mathbf{v}}$
  $\mathbf{W}_{ij} = \mathbf{J}_{\mathbf{C}_{ij}}^{\top} \mathbf{J}_{\mathbf{X}_{ij}}$
  Mark $\mathbf{V}_{jj}$ updated
end for

Step2: Schur Complement

- Batch BA

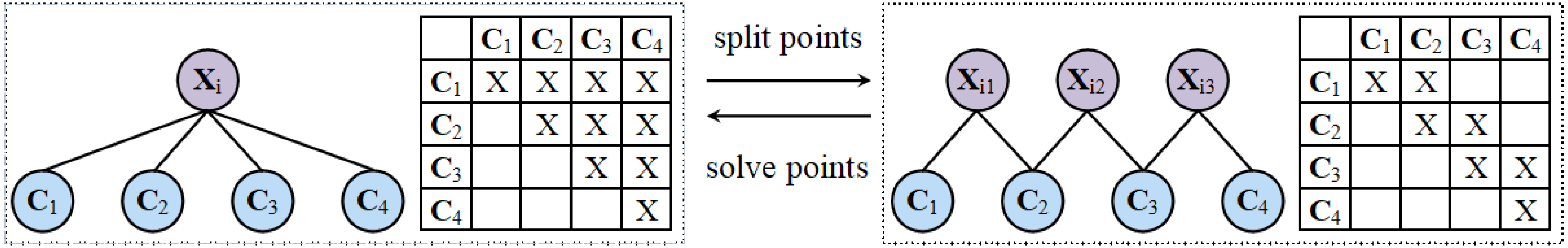
$\mathbf{S} = \mathbf{U}$
for each point j and each camera pair $(i_1, i_2) \in \mathcal{V}_j \times \mathcal{V}_j$ **do**
  $\mathbf{S}_{i_1 i_2} -= \mathbf{W}_{i_1 j} \mathbf{V}_{jj}^{-1} \mathbf{W}_{i_2 j}^{\top}$
end for
$\mathbf{g} = \mathbf{u}$
for each point j and each camera $i \in \mathcal{V}_j$ **do**
  $\mathbf{g}_i -= \mathbf{W}_{ij} \mathbf{V}_{jj}^{-1} \mathbf{v}_j$
end for

- ICE-BA

for each point j that $\mathbf{V}_{jj}$ is updated and each camera pair $(i_1, i_2) \in \mathcal{V}_j \times \mathcal{V}_j$ **do**
  $\mathbf{S}_{i_1 i_2} += \mathbf{A}^{\mathbf{S}}_{i_1 i_2 j}$
  $\mathbf{A}^{\mathbf{S}}_{i_1 i_2 j} = \mathbf{W}_{i_1 j} \mathbf{V}_{jj}^{-1} \mathbf{W}_{i_2 j}^{\top}$
  $\mathbf{S}_{i_1 i_2} -= \mathbf{A}^{\mathbf{S}}_{i_1 i_2 j}$
end for
for each point j that $\mathbf{V}_{jj}$ is updated and each camera $i \in \mathcal{V}_j$ **do**
  $\mathbf{g}_i += \mathbf{b}^{\mathbf{g}}_{ij}$; $\mathbf{b}^{\mathbf{g}}_{ij} = \mathbf{W}_{ij} \mathbf{V}_{jj}^{-1} \mathbf{v}_j$; $\mathbf{g}_i -= \mathbf{b}^{\mathbf{g}}_{ij}$
end for

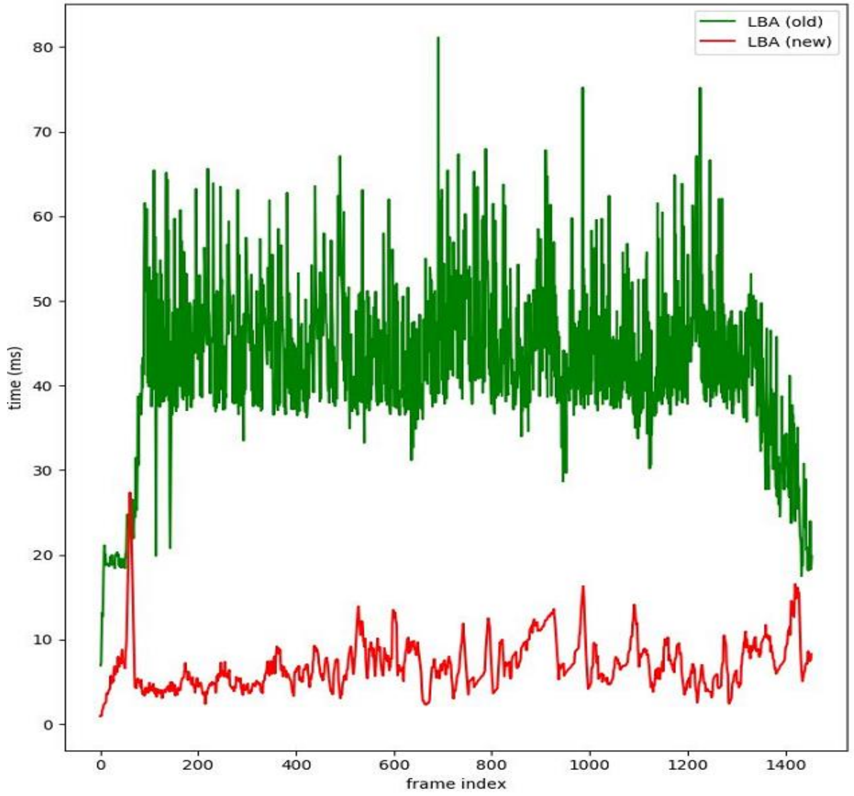
Sub-track Improvement for Local BA

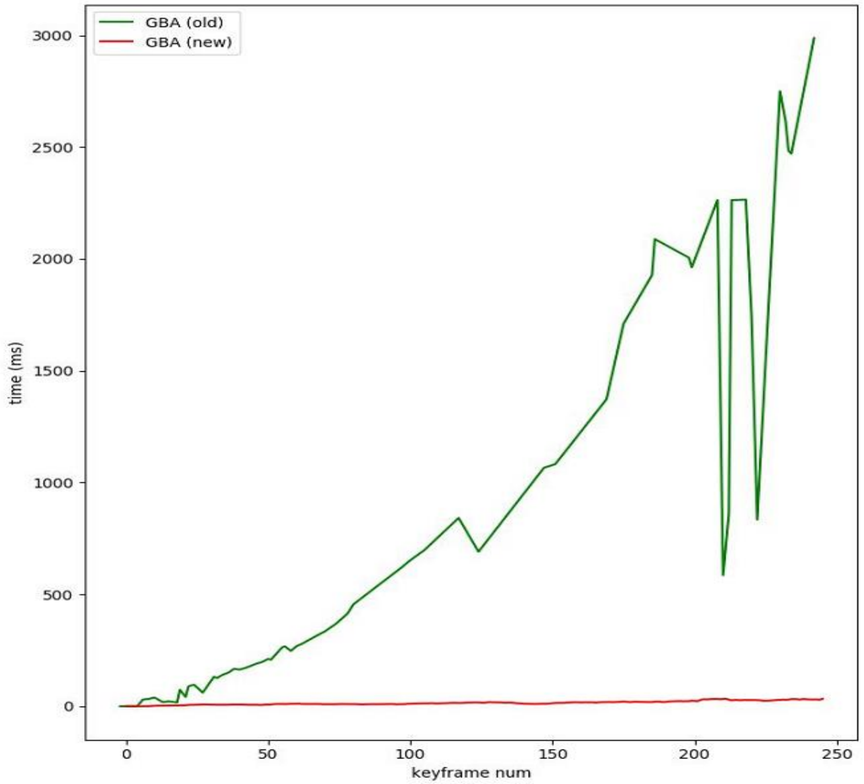
- In Local BA, most points may be observed by most frames in the sliding window
  - Dense Schur complement
  - A large portion need to be re-computed
- Split the original long feature track X_i into several short overlapping sub-tracks $X_{i_1}, X_{i_2}, \cdots$



Efficiency Comparison

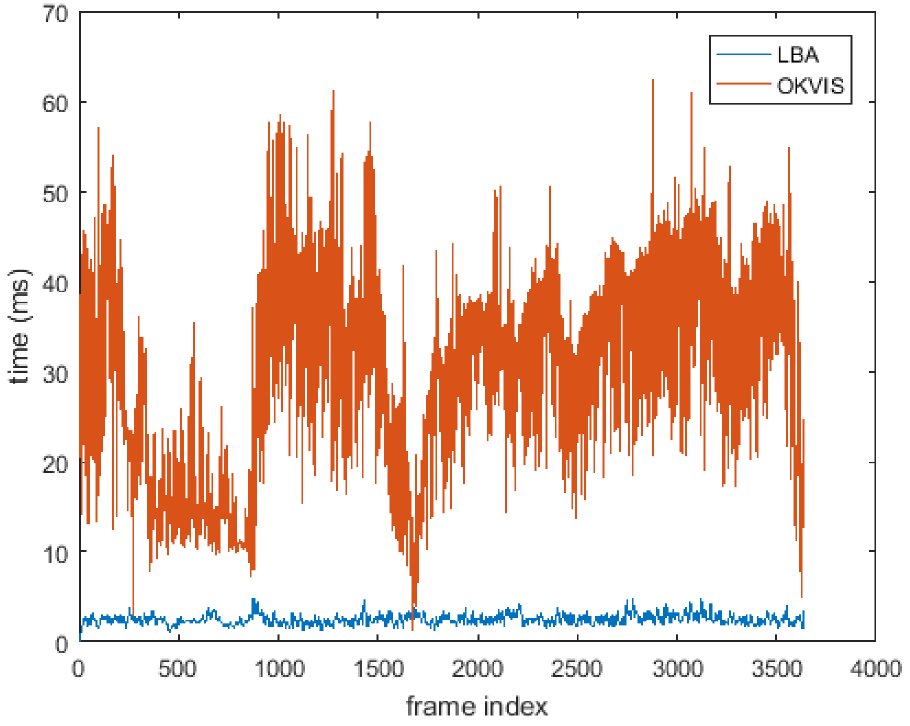
- Local BA (LBA)
  - ICE-BA (50 frames)
  - Ceres (10 key frames)
- Global BA (GBA)
  - ICE-BA: almost $O(1)$
  - Ceres: $O(n^2)$



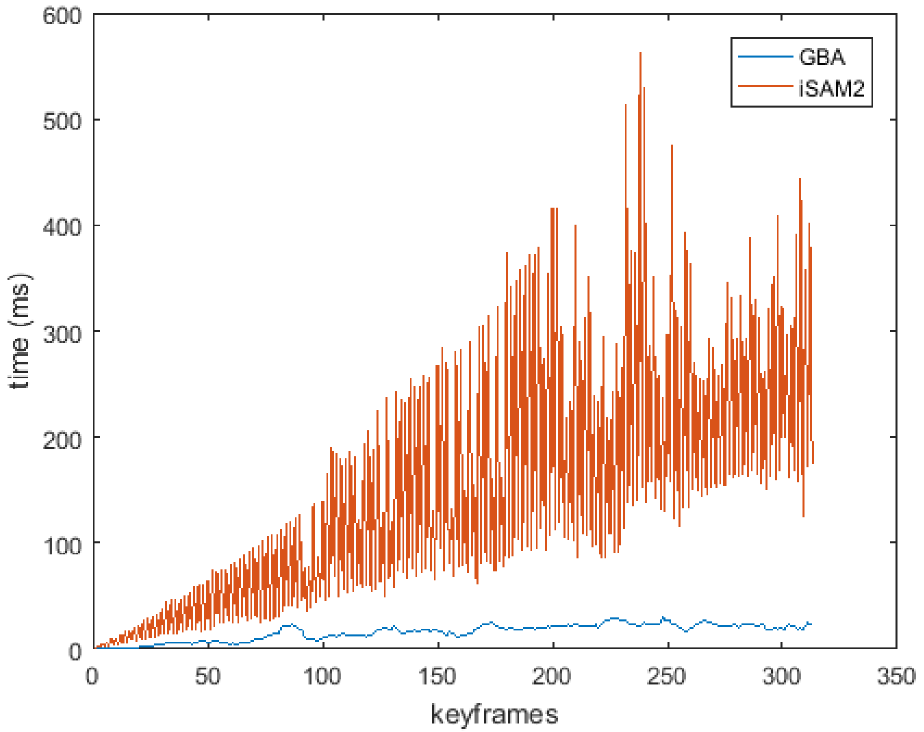


Efficiency Comparison

- Local BA (LBA)
  - ICE-BA (50 frames)
  - OKVIS (8 key frames)
  - 10x speedup



- Global BA (GBA)
  - ICE-BA: steady and smooth
  - iSAM2: steep and peaks
  - 20x speedup



Accuracy Comparison on EuRoc dataset

Seq.	Ours w/ loop	Ours w/o loop	OKVIS	SVO	iSAM2
MH_01	0.11	0.09	0.22	**0.06**	0.07
MH_02	0.08	**0.07**	0.16	0.08	0.11
MH_03	**0.05**	0.11	0.12	0.16	0.12
MH_04	**0.13**	0.16	0.18	-	0.16
MH_05	**0.11**	0.27	0.29	0.63	0.25
V1_01	0.07	0.05	**0.03**	0.06	0.07
V1_02	0.08	**0.05**	0.06	0.12	0.08
V1_03	**0.06**	0.11	0.12	0.21	0.12
V2_01	0.06	0.12	**0.05**	0.22	0.10
V2_02	**0.04**	0.09	0.07	0.16	0.13
V2_03	**0.11**	0.17	0.14	-	0.20
Avg	**0.08**	0.12	0.14	0.20	0.13

Comparison and Analysis

	iSAM2	EIBA / ICE-BA
Motion style	✓ keep forward ✗ to and fro	Suitable for any motion
Variable ordering	By algebraic method - The best ordering are changed time to time	By tricks of standard BA - Don't care about which camera/point comes first - Always marginalize points first - PCG explicitly leverage sparsity of camera Hessian
Incremental calculation vs sparseness	Trade off - Fix linearization: matrix becomes denser and denser during to and fro motion. - Reordering: re-calculation	Re-linearize wherever necessary without affecting sparseness

Source Code: https://github.com/baidu/ICE-BA

Key Idea： Cloud-Edge-Terminal Combination

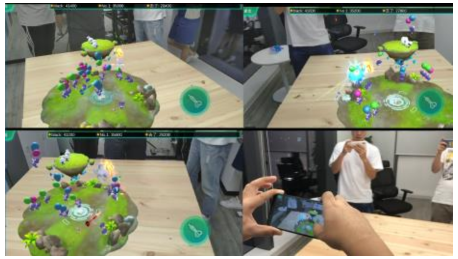
Localization & Navigation

Apply

Terminal

Apply

Multiple-Persons Sharing



5G
High Bandwith, Low Latency

Optimization results
Pose information
3D map
Real-time rendering & feedback

Sensor Data

Cloud+Edge Computing

3D Registration

- **Multi-sensor fusion SLAM/semantic SLAM**
- Object Recognition& Tracking
- 3D Reconstruction of Scenes

Large-Scale AR

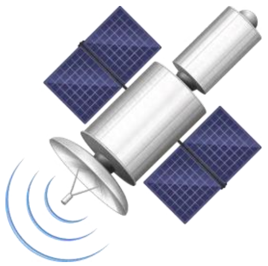
- Light Estimation
- Realistic Rendering
- Occlusions Handling
- Physical Simulation

HD Map

Visual Localization & AR Navigation

Localization & AR Navigation

Traditional Solutions



GPS

- Error up to 10 meters
- Not available in indoor environments

WIFI, Blue Tooth

- Additional deployment
- Expensive hardware

Visual Solutions

Advantages

- Low cost
- Non-intrusive
- High precision
- Intuitive with AR effect

Challenges

- Lack of visual features
- Environment change
- Heavy computation

Main Techniques in Visual Localization & AR Navigation

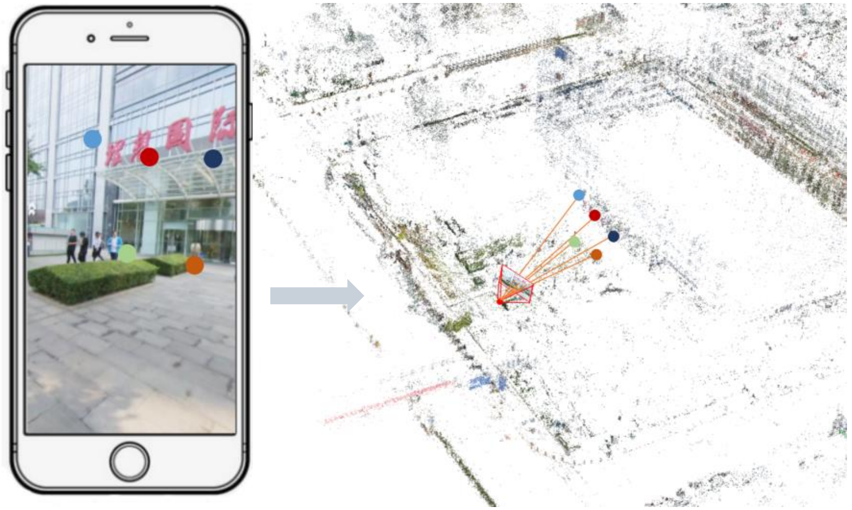
Sparse Map Reconstruction

- Extract visual features
- Recover the 3D structure

Dense Map Reconstruction

- Handling occlusions and collisions
- Free-viewpoint 3D navigation

Visual Localization & Tracking

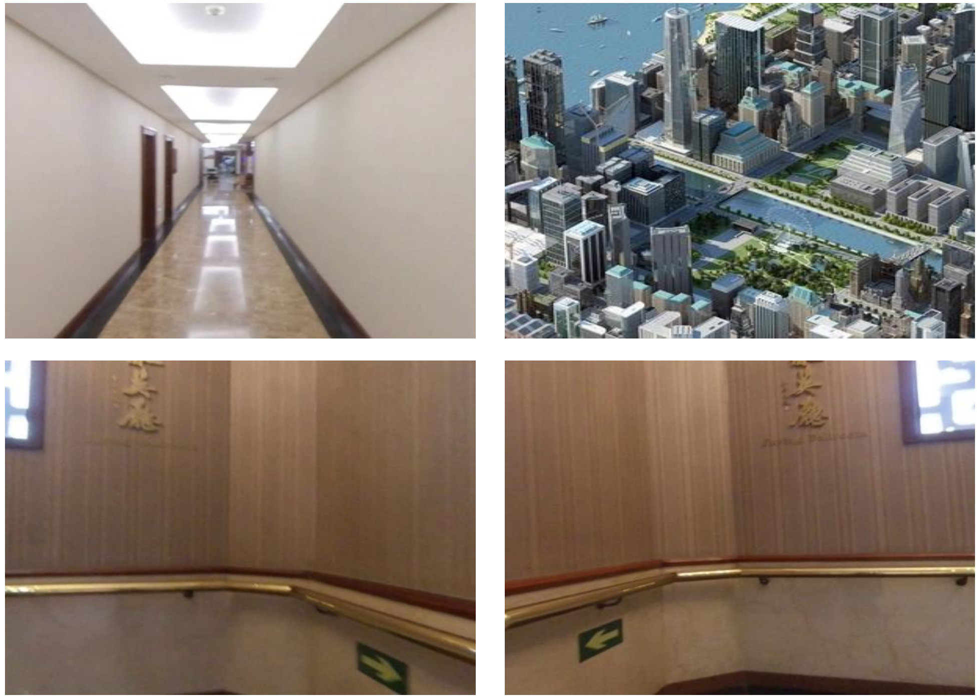


- Real-time 6DOF camera pose recovery for AR

Sparse Map Reconstruction

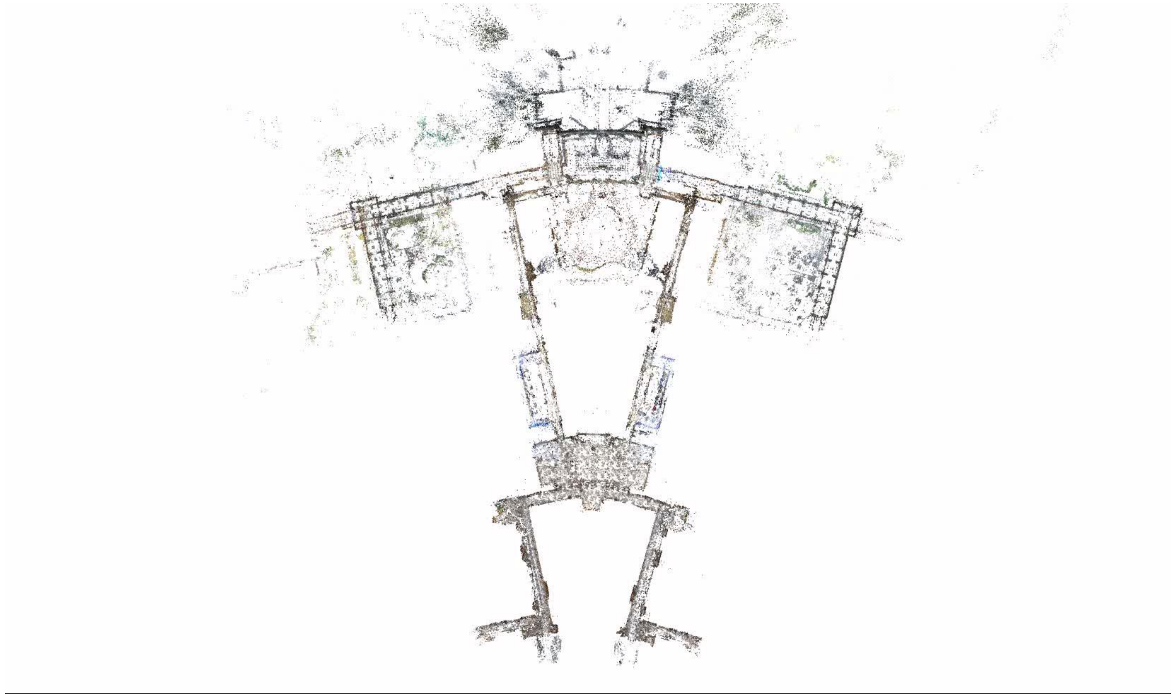
Challenges

- Many textureless regions
- Visual Ambiguity
- Large Scale



Key Ideas

- Capture Panorama Videos
- Integrating SLAM with SfM
- Divide and conquer



Dense Map Reconstruction

Challenges

- Many textureless regions
- Large-scale reconstruction

Key Ideas

- Accurate dense depth maps estimation and fusion
  - By multi-level feature matching
- Extendable accurate dense mesh reconstruction
  - Out-of-core reconstruction of large-scale meshes

Visual Localization & Tracking

Loose coupling

Tight coupling

Challenges

- Real-time
- Long distance
- View point, illumination, appearance variations

Key Ideas

- Cloud and terminal cooperation
- Tightly couple SLAM with global relocalization
- Learning based visual features

AR Multiple-Persons Sharing

Reconstructed 3D Map

Softwares

- ENFT-SFM or LS-ACTS
  - http://www.zjucvg.net/ls-acts/ls-acts.html
- RKSLAM: http://www.zjucvg.net/rkslam/rkslam.html
- RDSLAM: http://www.zjucvg.net/rdslam/rdslam.html
- ACTS: http://www.zjucvg.net/acts/acts.html
- SenseSLAM
  - http://www.zjucvg.net/senseslam/
  - http://openar.sensetime.com/

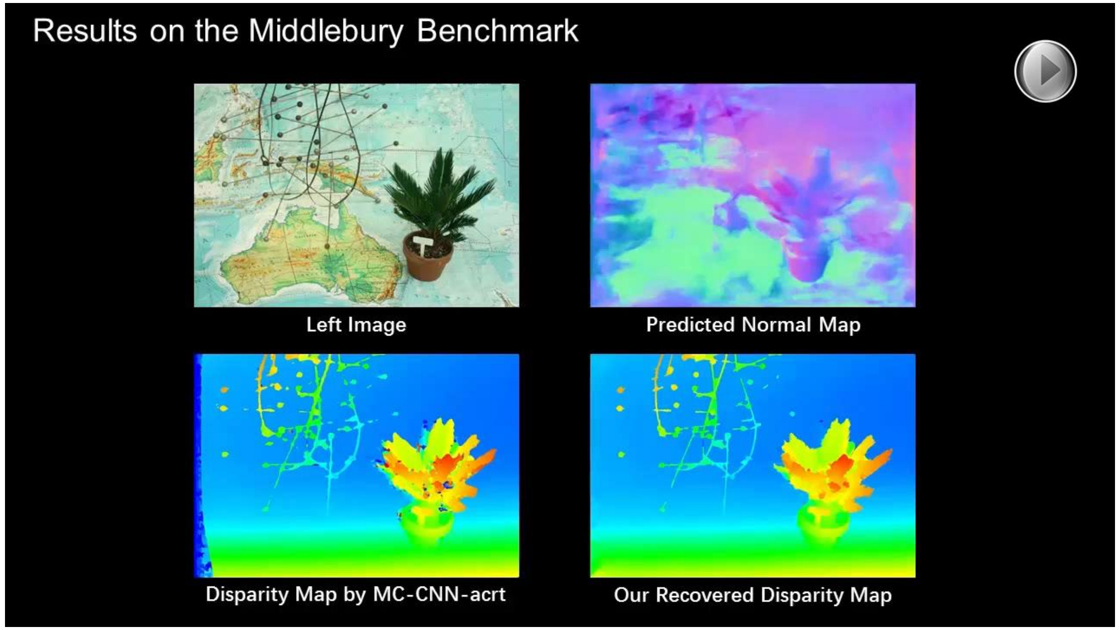
Source Code

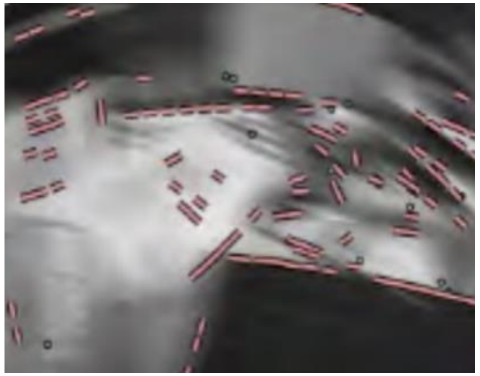
- ENFT-SfM
  - https://github.com/zju3dv/ENFT-SfM
- Segment-based Bundle Adjustment
  - https://github.com/zju3dv/SegmentBA
- Efficient Incremental Bundle Adjustment
  - EIBA: https://github.com/zju3dv/EIBA
  - ICE-BA: https://github.com/baidu/ICE-BA

VSLAM/VISLAM Technology Trends (1)

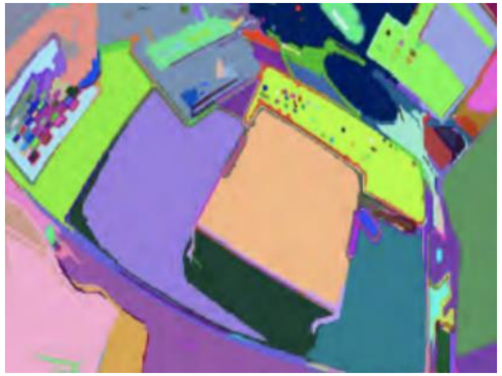
- **Reduce Textureless Problem**
  - Edge Tracking
  - Direct Tracking
  - Learning based methods or incorporating scene prior/semantic information

  (predict scene layout/semantic information, depth map and normal map)

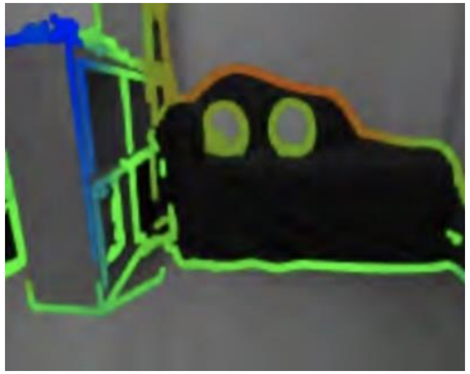




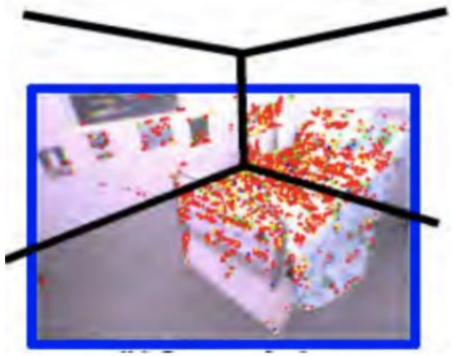
Edge Feature
(Klein et al., 2008)



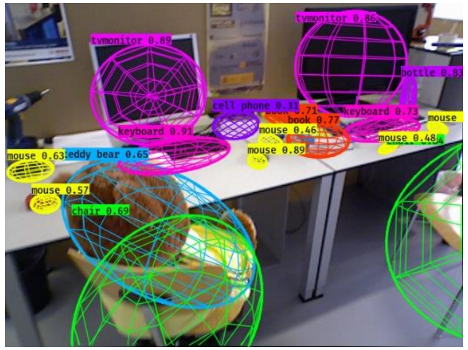
Plane Feature
(Concha et al., 2014)



Semi-dene Tracking
(Engel et al., 2014)



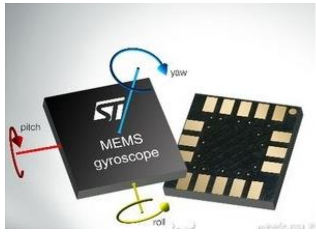
Scene Layout
(Salas et al., 2015)

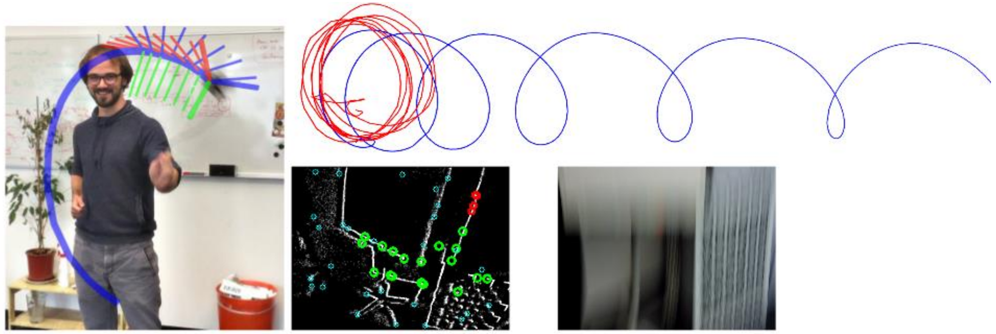


Semantic SLAM
(Nicholson et al., 2018)

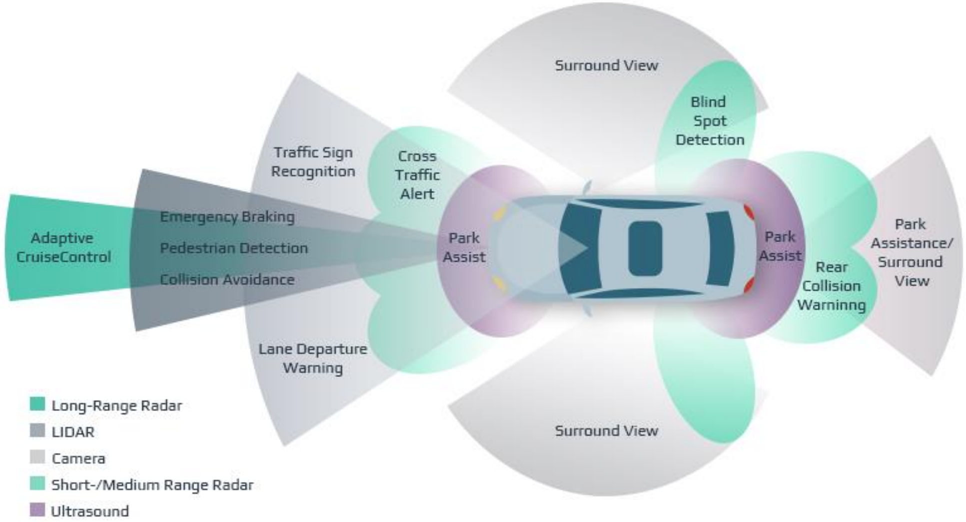
VSLAM/VISLAM Technology Trends (2)

- **Multiple Sensors Fusion**
  - Combining GPS, depth camera, odometer， WiFi， 5G

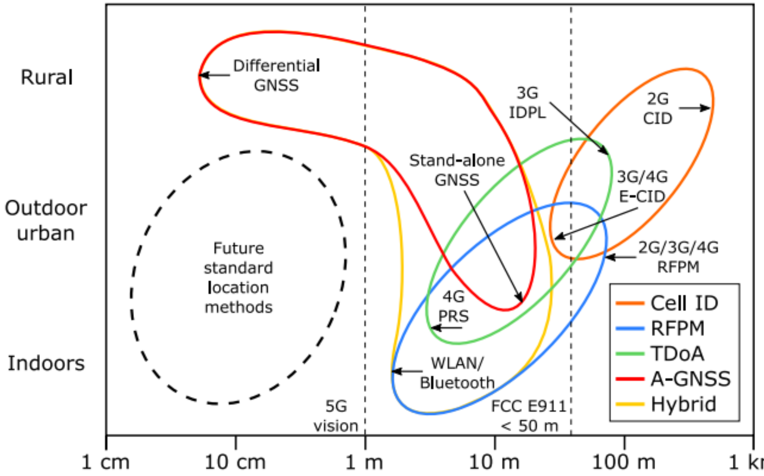




Even-camera based VIO（Rebecq et al., 2017）



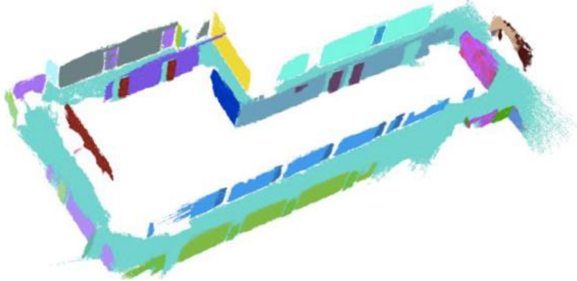
Multiple Sensors Fusion
https://www.intellias.com/sensor-fusion-autonomous-cars-helps-avoid-deaths-road/



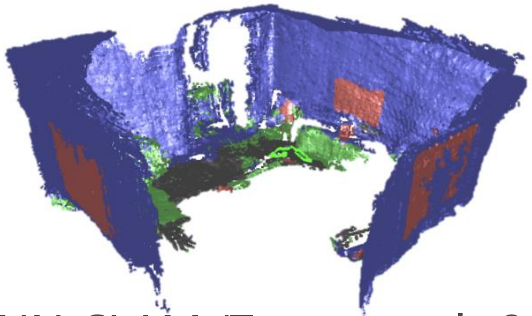
Survey of cellular mobile radio localization methods
（Peral-Rosado et al., 2017）

VSLAM/VISLAM Technology Trends (3)

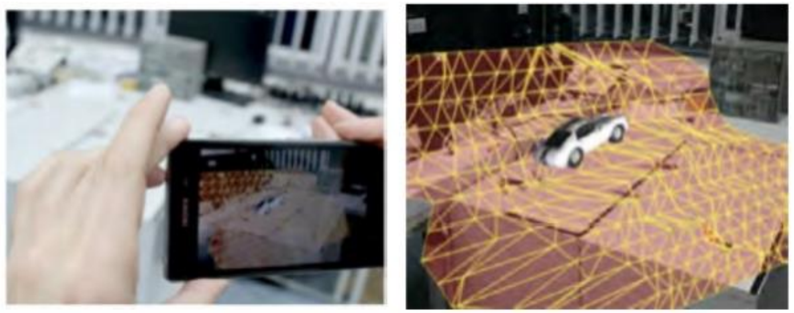
- **Dense 3D Reconstruction**
  - Real-time singe / multiple camera based methods
  - Real-time depth camera based methods
  - Real-time reconstruction of non-rigid objects



Keyframe-based Dense Planar SLAM
(Hsiao et al., 2017)

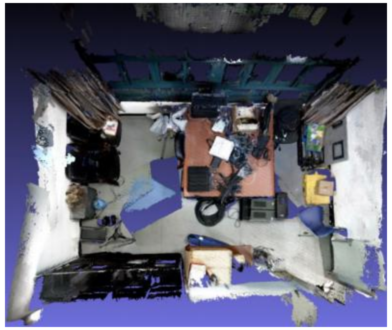


CNN-SLAM (Tateno et al., 2017)



Dense 3D Reconstruction based
AR application
(Schöps et al., 2014)

MobileFusion
(Ondrúška et al., 2015)



RKD-SLAM
(Liu et al., 2017)

DynamicFusion
(Newcombe et al., 2015)

Thanks！