ORB SLAM 2 : an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras

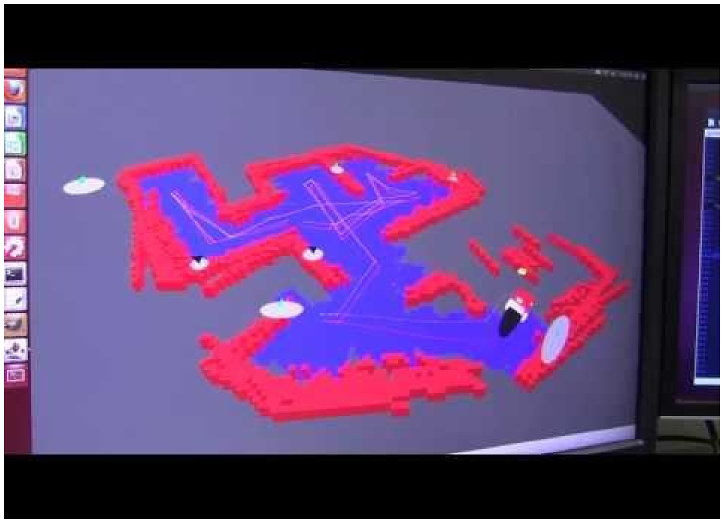
Raul ur-Artal and Juan D. Tardos

Presented by:
Xiaoyu Zhou
Bolun Zhang
Akshaya Purohit
Lenord Melvix

Outline

- **Background**
- Introduction
- Tracking
- Local mapping
- Loop closing
- Experiments and Results

Motivation



3

What is SLAM ?

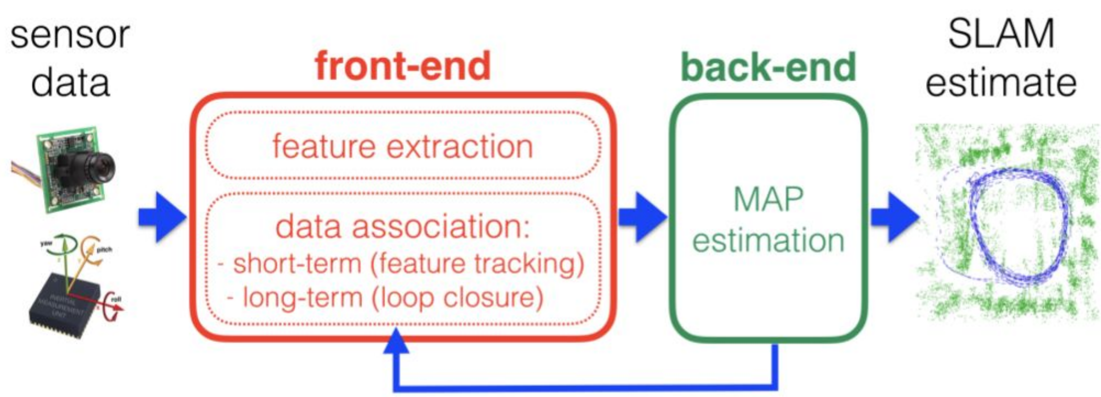
- Simultaneous localization and mapping

Why SLAM ?

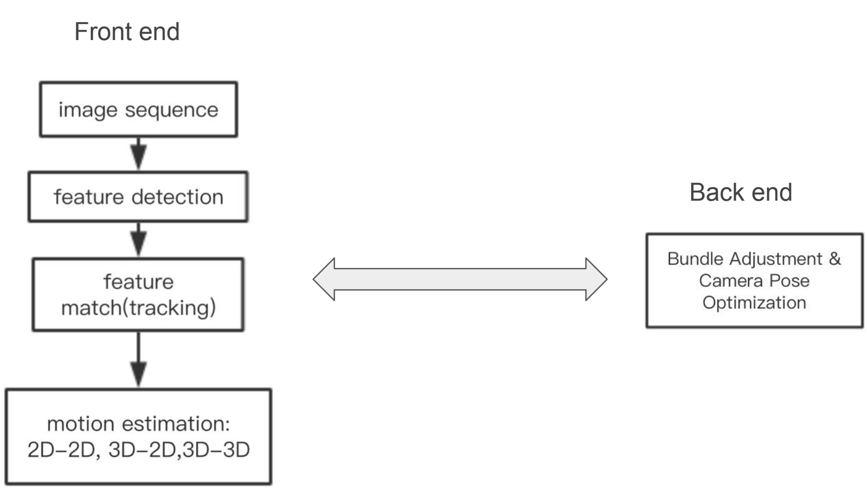
- In an environment without GPS, how is localization achieved?

4

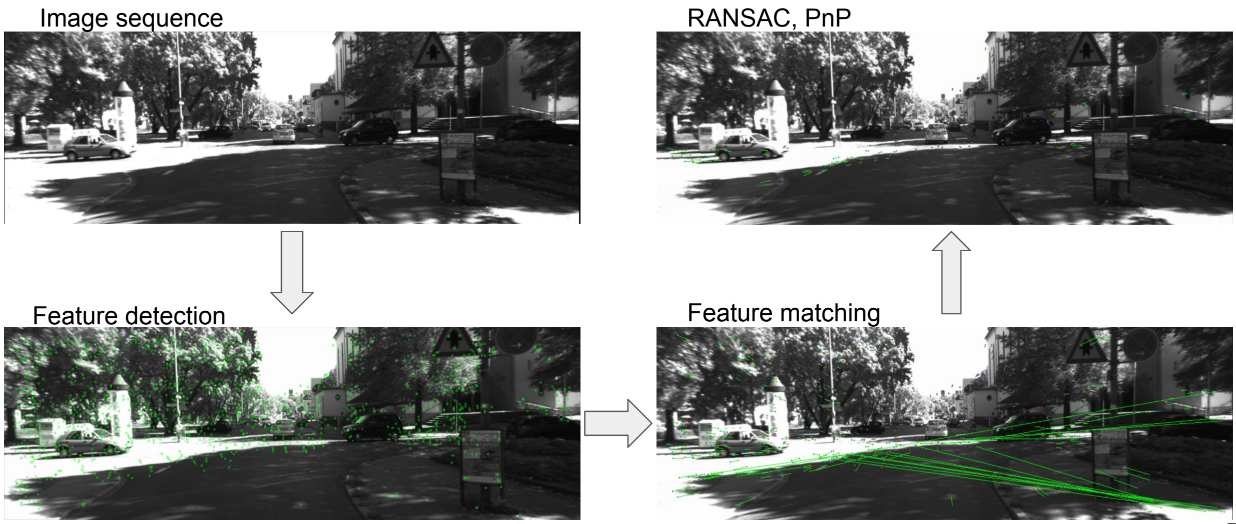
Visual SLAM: Main Parts



Visual SLAM: Front-End, Back-End flow chart



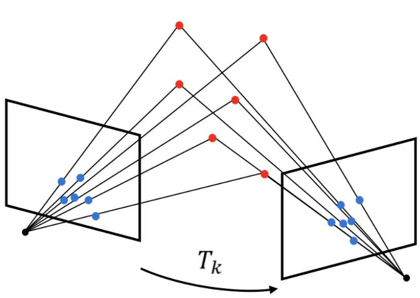
Visual SLAM: Front-End flow chart

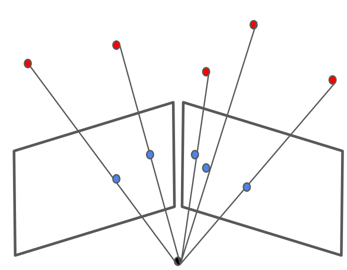


Visual SLAM: Front-End

Motion estimation: **2D-2D: Essential Matrix, Planar Projective Transformation Matrix**

- minimize reprojection error
- Impossible if the camera purely rotates



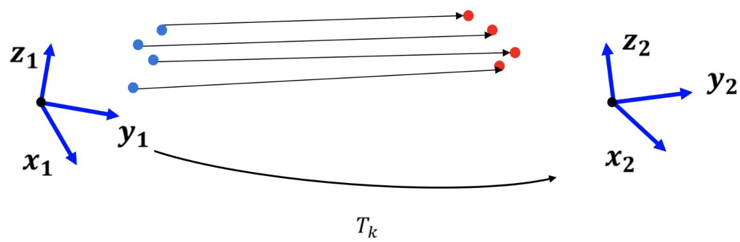


Visual SLAM: Front-End

Motion estimation: **3D-3D: Iterative Closest Point (ICP)**

Given two sets of 3D points, iteratively estimate the transformation Tk that can minimize the 3D-3D distance.

$$T_k = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix} = \arg\min_{T_k} \sum_i ||\tilde{X}_k^i - T_k \tilde{X}_{k-1}^i||$$

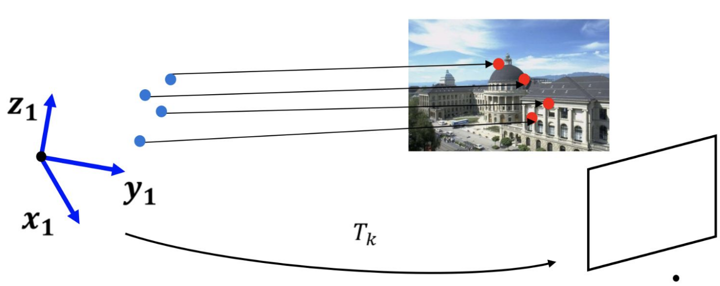


Visual SLAM: Front-End

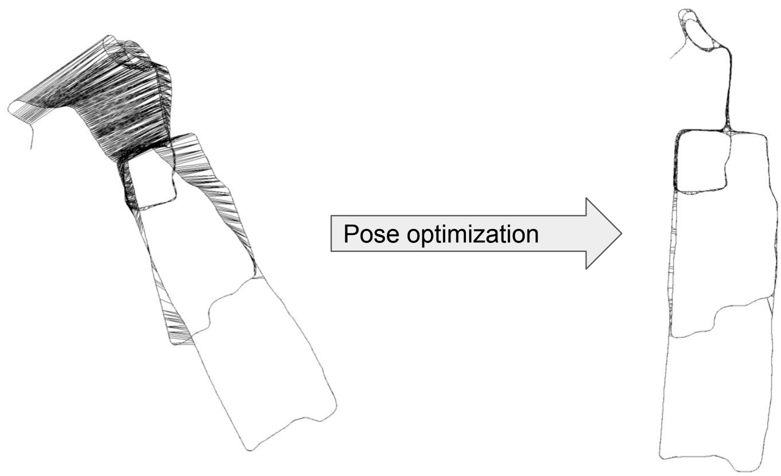
Motion estimation: **3D-2D: Perspective from n Points (PnP)**

The solution is found by determining the transformation that minimizes the reprojection error.

$$T_k = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix} = \arg\min_{T_k} \sum_i ||p_k^i - \hat{p}_{k-1}^i||^2$$



Visual SLAM: Back-End flow chart

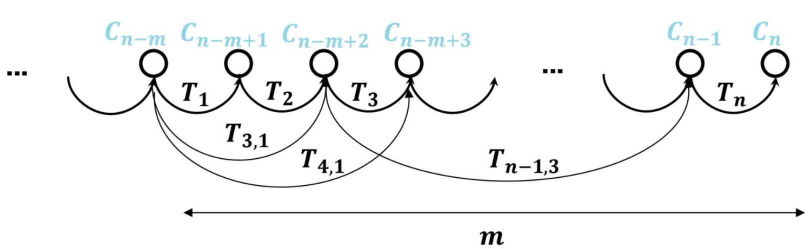


Visual SLAM: Back-End

Camera Pose optimization:

- Each node represents a pose of the camera
- Each edge represents a constraint between two nodes.
- Minimize function below to improve camera's poses.

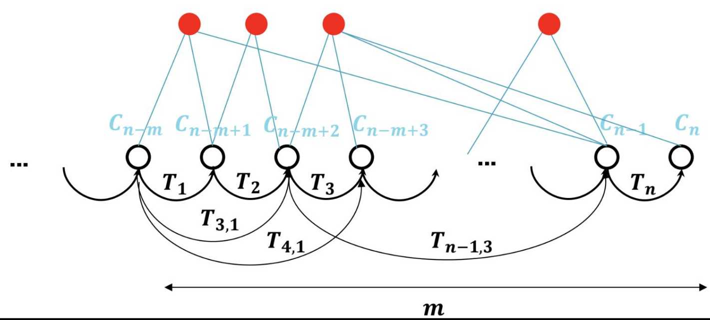
$$\sum_{e_{ij}} \|C_i - T_{e_{ij}} C_j\|^2$$



Visual SLAM: Back-End

Bundle Adjustment(BA):

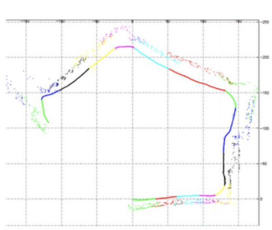
- Very similar to camera-pose optimization,
- Also optimize the position of 3D points, minimize reprojection error.
- Extremely time consuming.



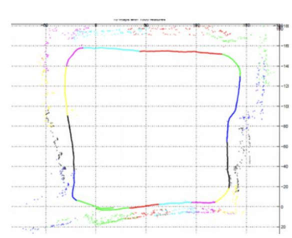
Visual SLAM: Strongest Constraint

Loop Closure:

- The most valuable constraint for pose-graph optimization.
- Usually between nodes that are far away, which may have large drift.
- Very afraid of false positive, which can destroy the entire map.



Before Loop Closure



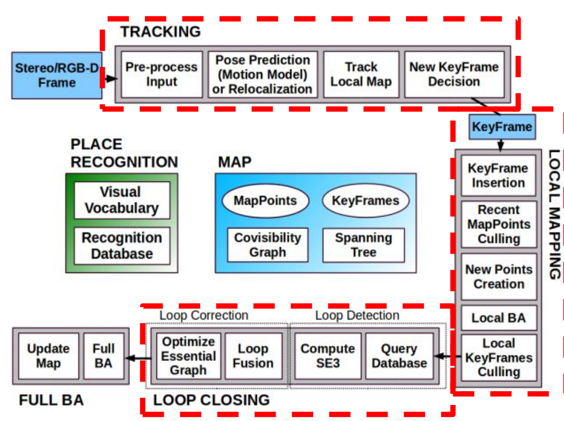
After Loop Closure

Outline

- Background
- **Introduction**
- Tracking
- Local mapping
- Loop closing
- Experiments and Results

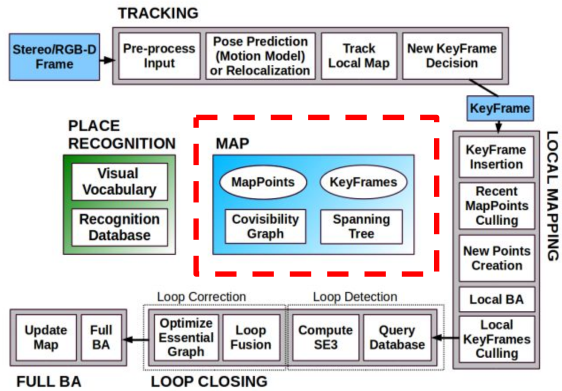
ORB-SLAM2: System Overview

- Feature-based
- Monocular, Stereo, and RGB-D
- Loop closing, relocalization and map reuse
- Three threads running in parallel
  - Tracking
  - Local Mapping
  - Loop Closing

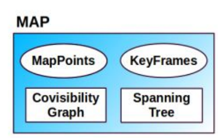


ORB-SLAM2: Map

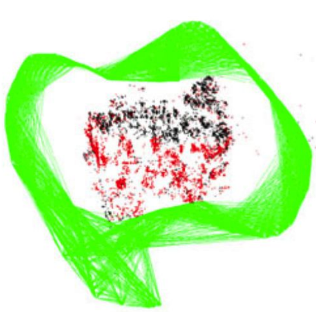
- **Map points**
  - 3D position
  - Viewing direction
  - Representative ORB descriptor
  - Viewing distance
- **Keyframes**
  - Camera pose
  - Camera intrinsics
  - ORB features in the frame



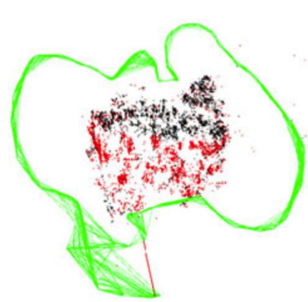
ORB-SLAM2: Map



- **Covisibility Graph**
  - Node: Keyframe
  - Edge: Share observations of map points
  - Min shared map points:15

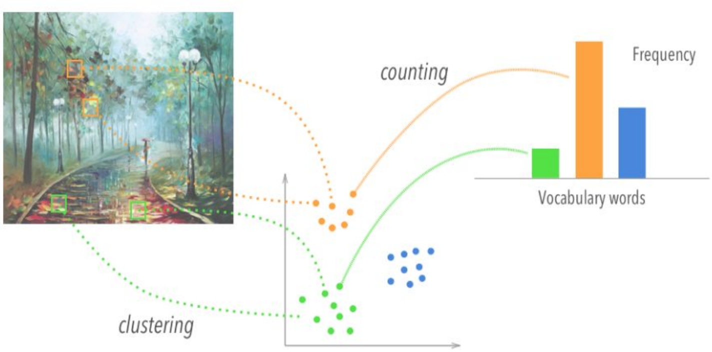


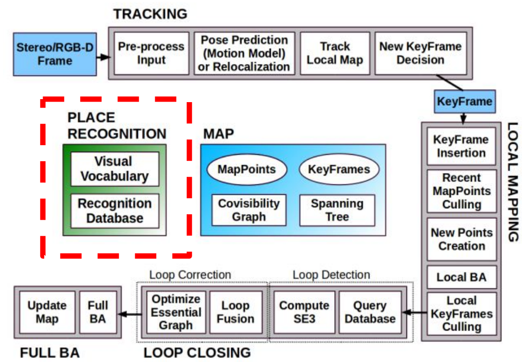
- **Essential Graph**
  - Subgraph of covisibility graph
  - Spanning tree, high weight edges, loop closure edges
  - Min shared map points:100



ORB-SLAM2: Place Recognition

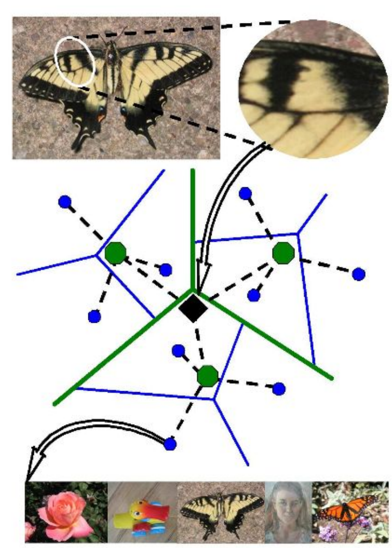
- **Visual Vocabulary**
  - Offline vocabulary of ORB descriptors extracted from a large set of images





ORB-SLAM2: Place Recognition

- **Recognition Database**
  - Database built incrementally, which stores for each visual word in the vocabulary, in which keyframes it has been seen.
  - Vocabulary tree using hierarchical k---means clustering
  - Leaves are the visual words

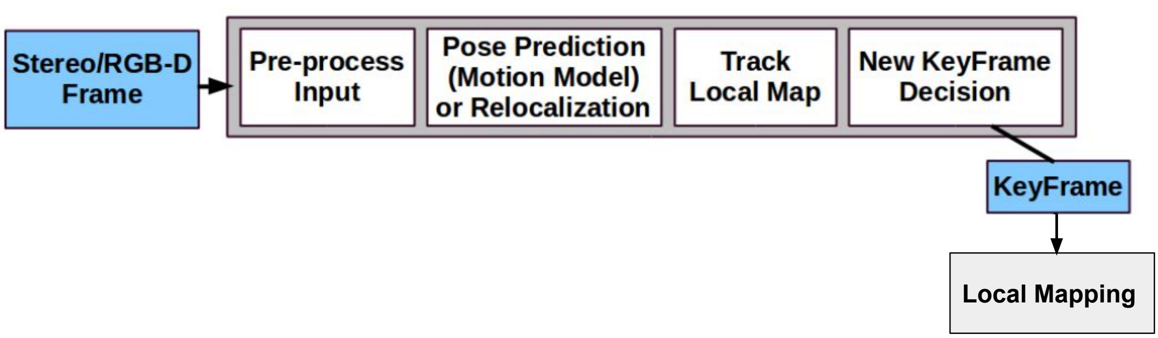


Outline

- Background
- Introduction
- **Tracking**
- Local mapping
- Loop closing
- Experiments and Results

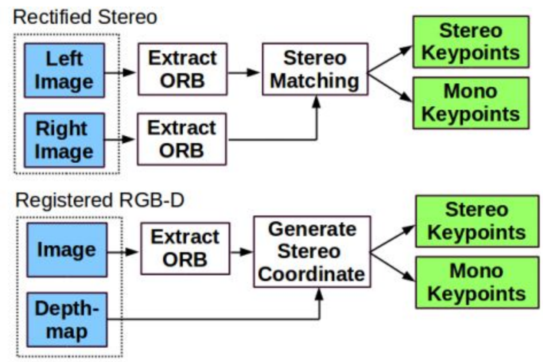
Tracking

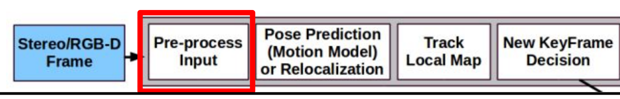
- Localize the camera with every frame and decide when to insert a new keyframe.



Tracking: Preprocess Input

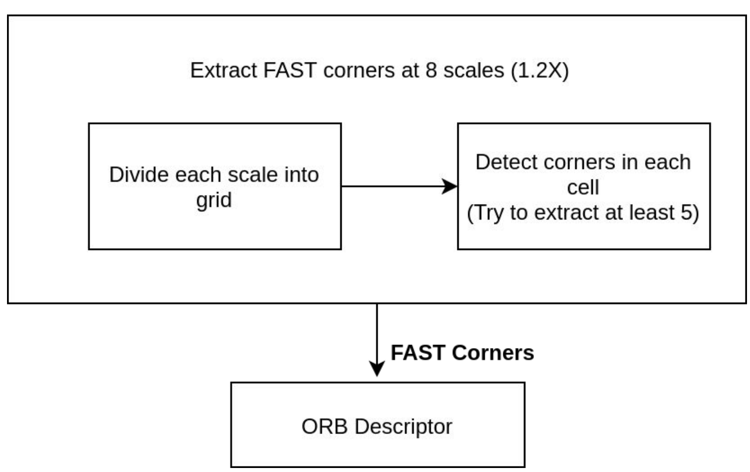
- Preprocess the input to extract features at salient keypoint locations
- All system operations are based on these features.
- Stereo Keypoints: (u_L, v_L, u_R)
  - Close: depth < 40X baseline
  - Far: Otherwise

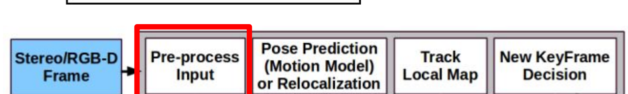




Tracking: Preprocess Input (Extract ORB)

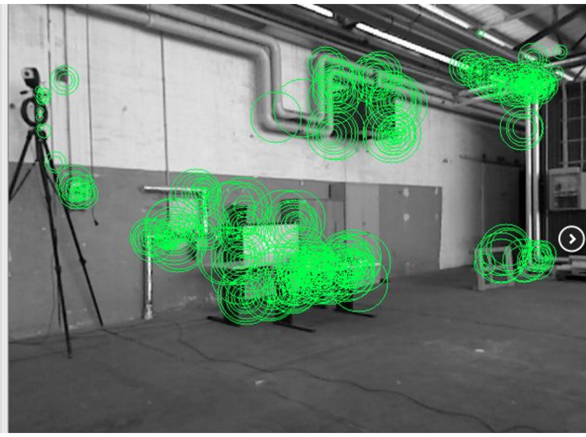
- Extremely fast to compute than SIFT or SURF





Tracking: Preprocess Input (Extract ORB)

ORB features in ORB-SLAM

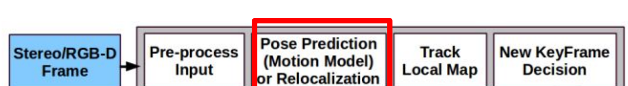


ORB features in general

Tracking: Pose Prediction or Relocalization

- **Pose Estimation From Previous Frame**
  - Constant velocity motion model to predict the camera pose
  - Perform a guided search.
  - Pose optimization

- **Pose Estimation via Global Relocalization (if tracking lost)**
  - Convert the frame into bag of words
  - Query the recognition database: Get matching Keyframes
  - Outlier rejection: RANSAC
  - PnP to get pose
  - Guided Search
  - Pose optimization

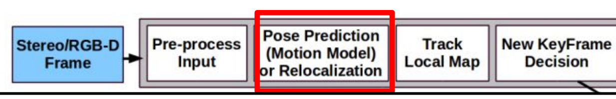


Tracking: Pose Prediction or Relocalization

- **Pose Optimization using Motion-only bundle adjustment:**
  - Optimize camera orientation **R** and position **t**
  - Minimizing error between matched 3D points in world coordinates and key points
  - Levenberg-Marquadt for non-linear optimization

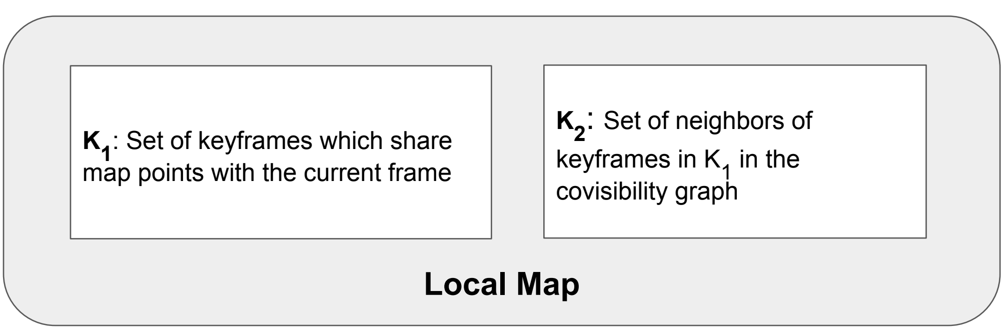
$$\{\mathbf{R}, \mathbf{t}\} = \underset{\mathbf{R},\mathbf{t}}{\operatorname{argmin}} \sum_{i \in \mathcal{X}} \rho \left(\left\| \mathbf{x}^i_{(\cdot)} - \pi_{(\cdot)} \left(\mathbf{R}\mathbf{X}^i + \mathbf{t} \right) \right\|^2_{\Sigma} \right)$$

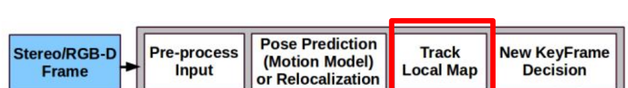
$$\pi_{\mathrm{m}}\left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}\right) = \begin{bmatrix} f_x \frac{X}{Z} + c_x \\ f_y \frac{Y}{Z} + c_y \end{bmatrix} \qquad \pi_{\mathrm{s}}\left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}\right) = \begin{bmatrix} f_x \frac{X}{Z} + c_x \\ f_y \frac{Y}{Z} + c_y \\ f_x \frac{X-b}{Z} + c_x \end{bmatrix}$$



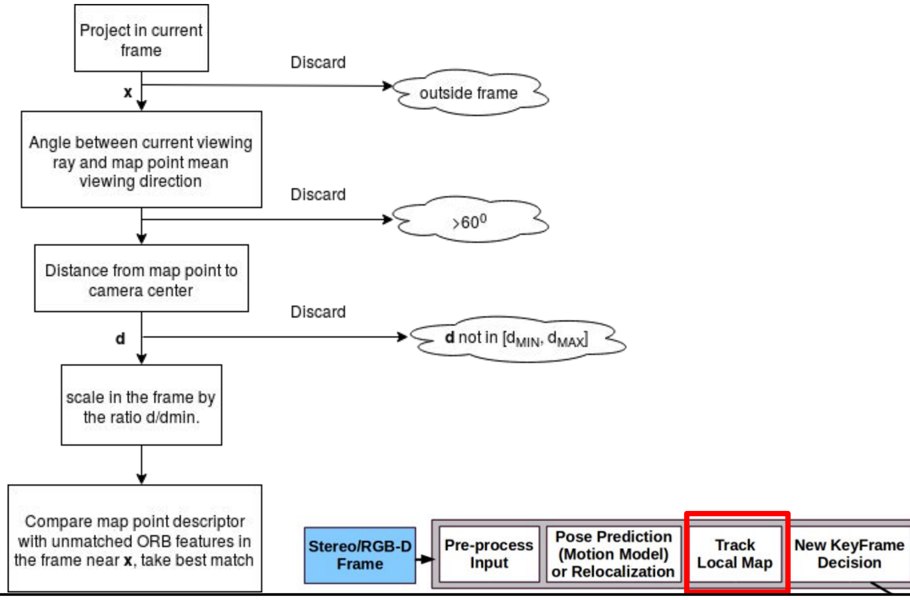
Tracking: Track Local Map

- Look into the local map for more map point correspondences.
- Pose optimization





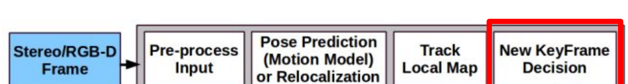
Tracking: Track Local Map



Tracking: New KeyFrame

Decision criteria (all required):

- ➢ More than 20 frames must have passed from the last global relocalization.
- ➢ Local mapping is idle, or more than 20 frames have passed from last keyframe insertion.
- ➢ Current frame tracks at least 50 points.
- ➢ Current frame tracks less than 90% points than Kref.

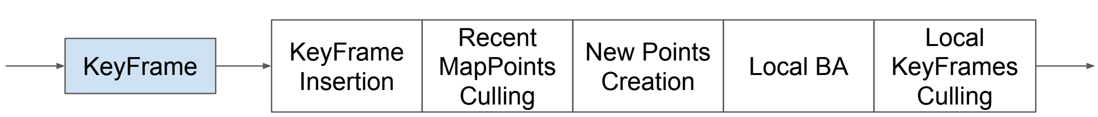


Outline

- Background
- Introduction
- Tracking
- **Local mapping**
- Loop closing
- Experiments and Results

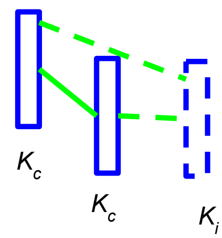
Local Mapping

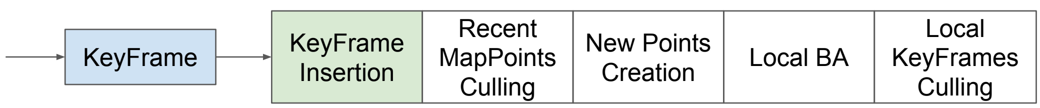
- Process new keyframes and performs local BA to optimize the map points and the poses of the keyframes



Local Mapping: Keyframe Insertion

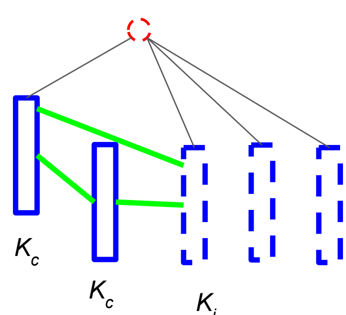
- Update the covisibility graph
  - Add new node and update edges
- Update the spanning tree in essential graph
  - Link with the keyframe with most shared points
- Compute the bags of words representation
  - Help triangulating new points

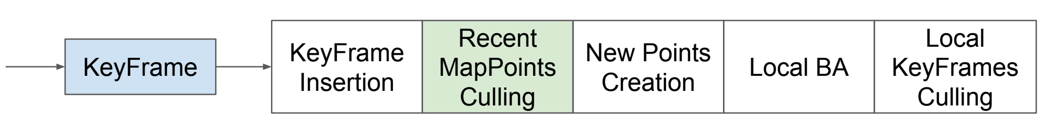




Local Mapping: Recent Map Points Culling

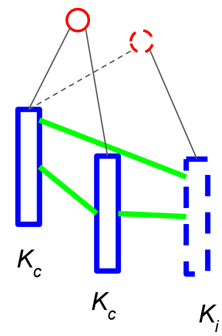
- Removal test after creation
  - Can be found in more than 25% of the predicted visible frames
  - Can be observed in at least three keyframes
- Keyframe culling
- Local BA discarding

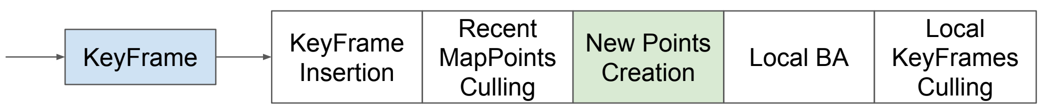




Local Mapping: New Map Point Creation

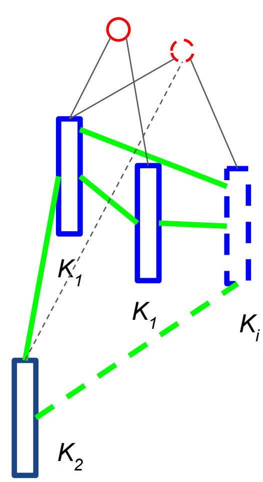
- New keyframe K_i and connected keyframes K_c in the covisibility graph
- For unmatched ORB in K_i, search match in K_c
  - Epipolar constraint
  - Speeds up by vocabulary tree
- Triangulate ORB pairs
  - Check depth, parallax, reprojection error, and scale consistency





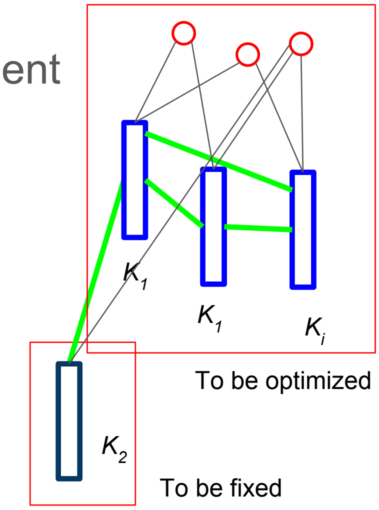
Local Mapping: New Map Point Creation

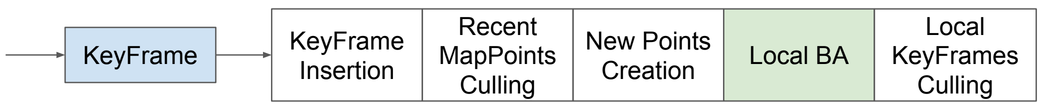
- Determine new map point properties
  - Mean unit vector of all its viewing directions
  - Representative descriptor
  - Observation distance
- Search correspondences in other keyframes
  - Connected keyframes K_1 in covisibility graph
  - Neighbor keyframes K_2 to the keyframes K_1
  - Project new map points to K_1 and K_2
  - Update covisibility graph



Local Mapping: Local Bundle Adjustment

- Optimize poses and map points
  - Current keyframe K_i
  - Connected keyframes K_c in the covisibility graph
  - Map points seen in K_i and K_c
- Fixed constraint
  - Keyframes with same map points but not connected to K_i
- Discard map points outliers and modify poses and map point coordinates





Local Mapping: Local Bundle Adjustment

- Optimizes set of co-visible keyframes and all points in those keyframes

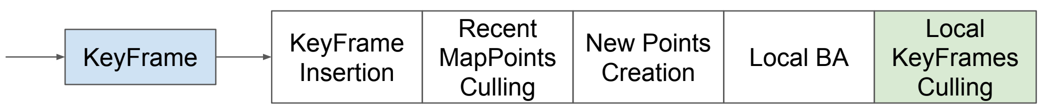
$$\{\mathbf{X}^i, \mathbf{R}_l, \mathbf{t}_l | i \in \mathcal{P}_L, l \in \mathcal{K}_L\} = \underset{\mathbf{X}^i, \mathbf{R}_l, \mathbf{t}_l}{\operatorname{argmin}} \sum_{k \in \mathcal{K}_L \cup \mathcal{K}_F} \sum_{j \in \mathcal{X}_k} \rho\left(E(k,j)\right)$$

$$E(k,j) = \left\| \mathbf{x}^j_{(\cdot)} - \pi_{(\cdot)}\left(\mathbf{R}_k \mathbf{X}^j + \mathbf{t}_k\right) \right\|^2_\Sigma$$

where K_L are set of co-visible keyframes, P_L are all points in those keyframes and K_F are other keyframes not in K_L observing points in P_L

Local Mapping: Local Keyframe Culling

- Reduce BA complexity and limit the number of keyframes
- Culling policy
  - Any keyframe in K_c whose 90% of the map points can be seen in at least three other keyframes

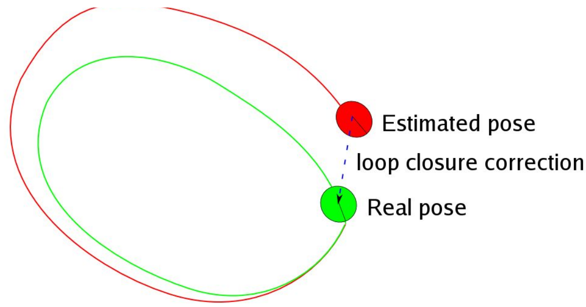


Outline

- Background
- Introduction
- Tracking
- Local mapping
- **Loop closing**
- Experiments and Results

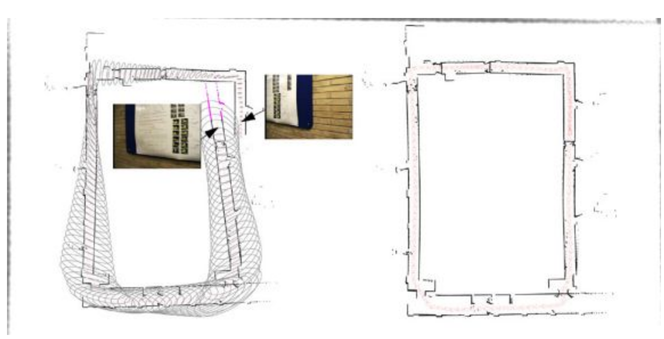
Loop Closing

Loop closing is the act of correctly asserting that a vehicle has returned to a previously visited location

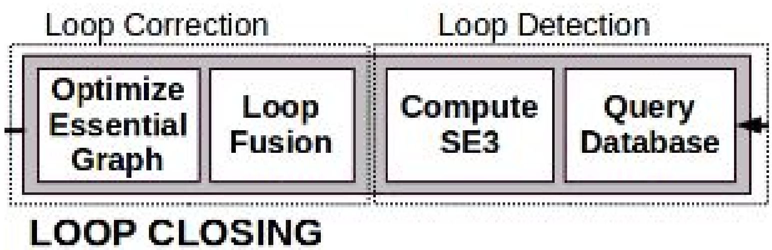


Why close loops?

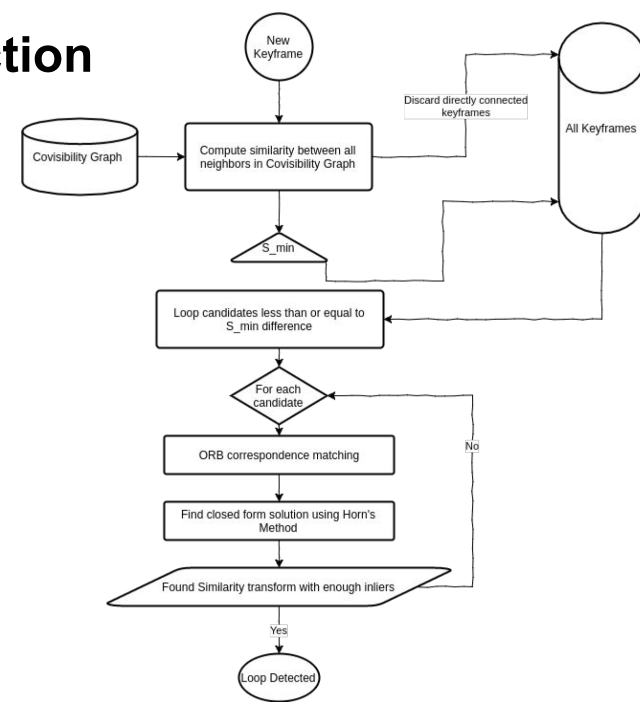
- Previously visited location gets remapped in wrong global location
- Error accumulates out-of-bound
- Incorrect loop detection is even more harder to recover.

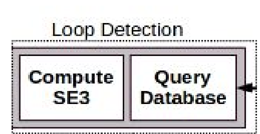


Loop Closing in ORB-SLAM2

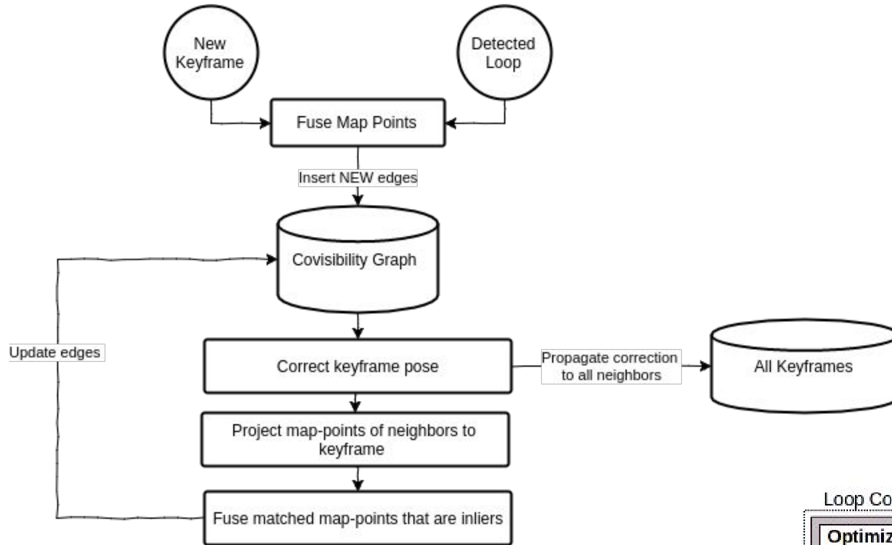


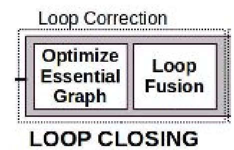
Loop Detection





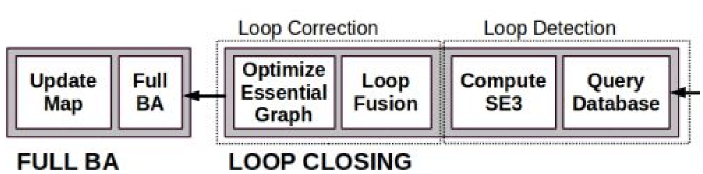
Loop Correction





Full Bundle Adjustment

- Optimize all KeyFrames and Points in the map
- Performed on separate thread after loop closure
- If new loop is detected, abort full BA and start again.



Outline

- Background
- Introduction
- Tracking
- Local mapping
- Loop closing
- **Experiments and Results**

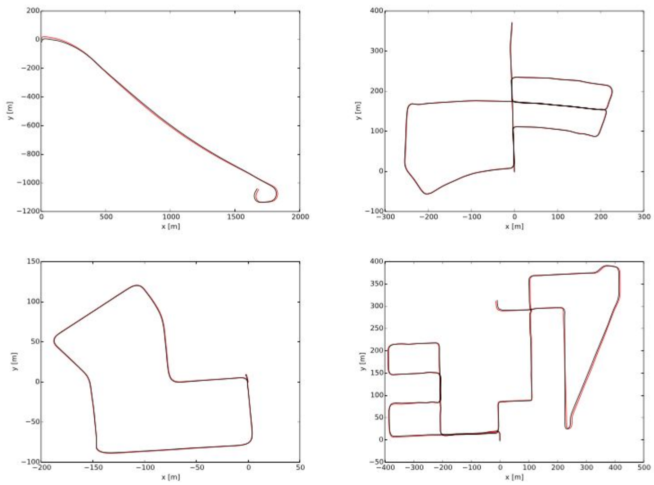
Experiments and Results

	ORB-SLAM2 (Stereo)			Stereo LSD-SLAM		
Error (Units)	t_{rel} (%)	r_{rel} (deg/100m)	t_{abs} (m)	t_{rel} (%)	r_{abs} (deg/100m)	t_{abs} (m)
00	0.70	**0.25**	1.3	**0.63**	0.26	**1.0**
01	**1.39**	**0.21**	10.4	2.36	0.36	**9.0**
02	**0.76**	0.23	5.7	0.79	0.23	**2.6**
03	**0.71**	**0.18**	**0.6**	1.01	0.28	1.2
04	0.48	**0.13**	0.2	**0.38**	0.31	0.2
05	**0.40**	**0.16**	**0.8**	0.64	0.18	1.5
06	**0.51**	**0.15**	**0.8**	0.71	0.18	1.3
07	**0.50**	**0.28**	0.5	0.56	0.29	0.5
08	**1.05**	0.32	**3.6**	1.11	**0.31**	3.9
09	**0.87**	0.27	**3.2**	1.14	**0.25**	5.6
10	**0.60**	**0.27**	**1.0**	0.72	0.33	1.5

KITTI dataset

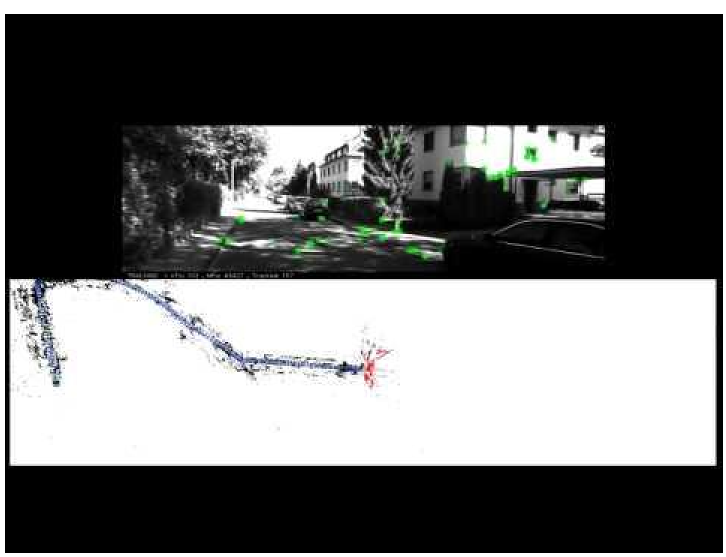
- Comparison with previously most successful open source stereo SLAM--LSD SLAM

Experiments and Results



- Generated camera trajectory compared with ground truth

ORB-SLAM



Thank You!!!