

# ORB-SLAM: A Versatile and Accurate Monocular SLAM System

16-833: Robot Localization and Mapping

**Sudharshan Suresh**

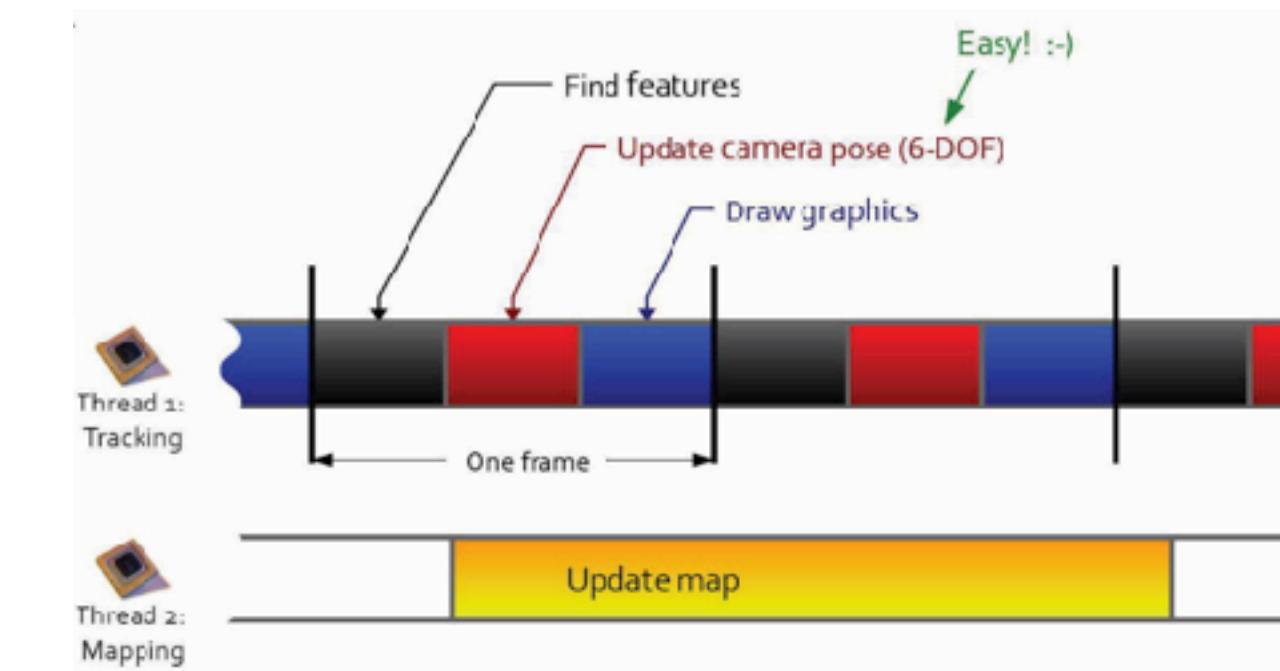
Fall 2019

Slides courtesy: Paloma Sodhi

# Last time: Parallel Tracking and Mapping

## Key Ideas

- Keyframe-based monocular SLAM system
- Splits mapping and tracking in two parallel threads
- Bundle Adjustment possible in real-time



## Limitations

- Restricted to small environments: no large-scale loop closing, keyframes grow unbounded
- Map initialization requires user interaction
- In tracking failure, re-localization not robust

# ORB-SLAM

Raúl Mur-Artal, J. M. M. Montiel and Juan D. Tardós

{raulmur, josemari, tardos} @unizar.es



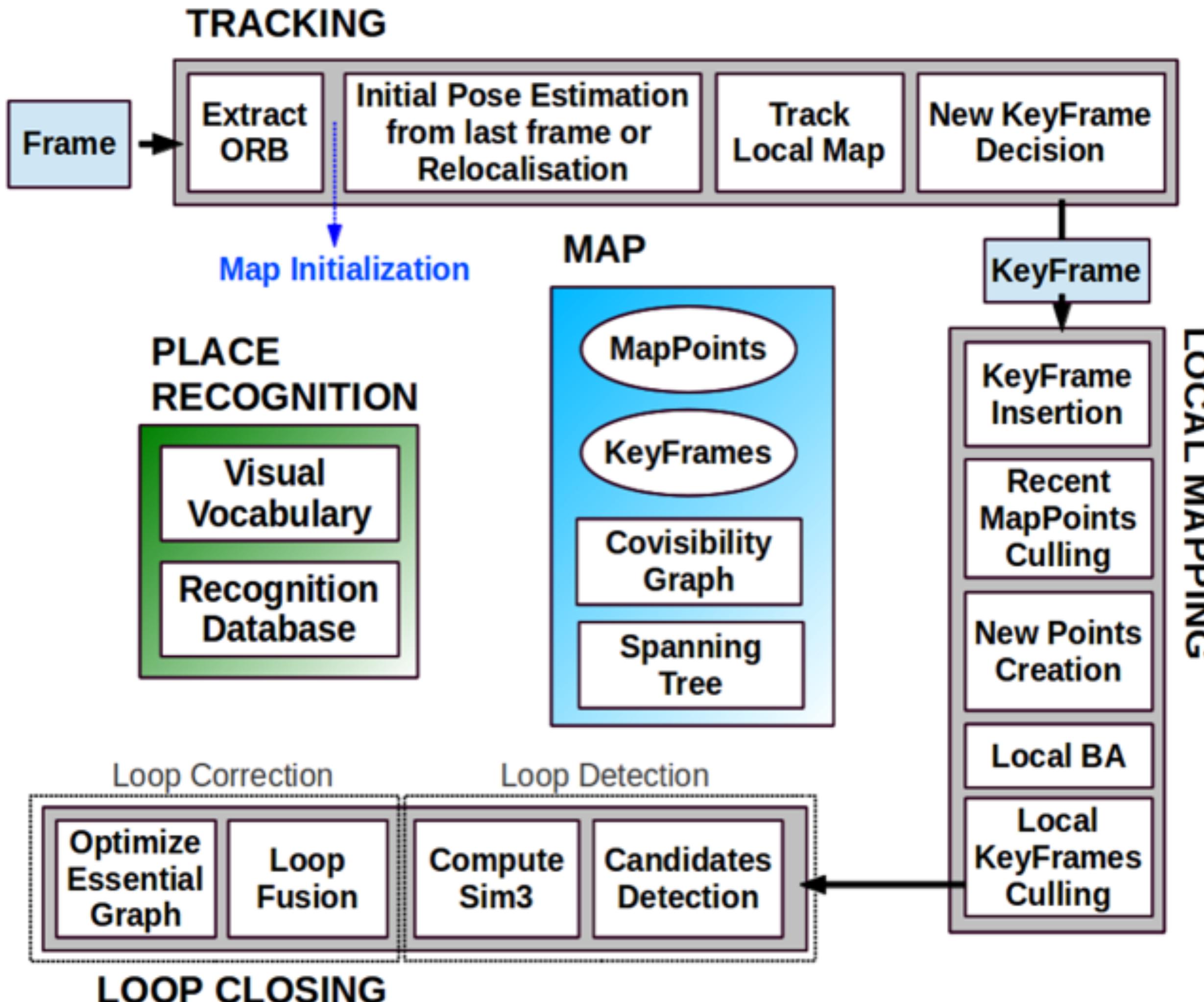
Instituto Universitario de Investigación  
en Ingeniería de Aragón  
**Universidad** Zaragoza



**Universidad**  
Zaragoza

# Threads: Tracking, Mapping, Loop Closing

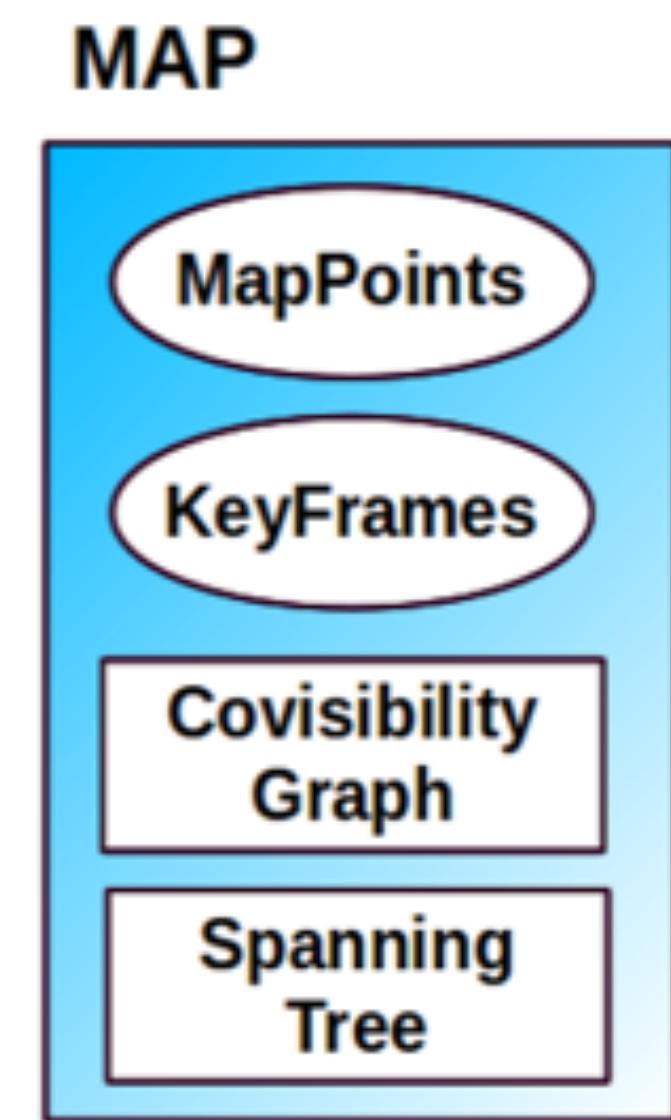
## Modules: Map, Place Recognition



# Outline

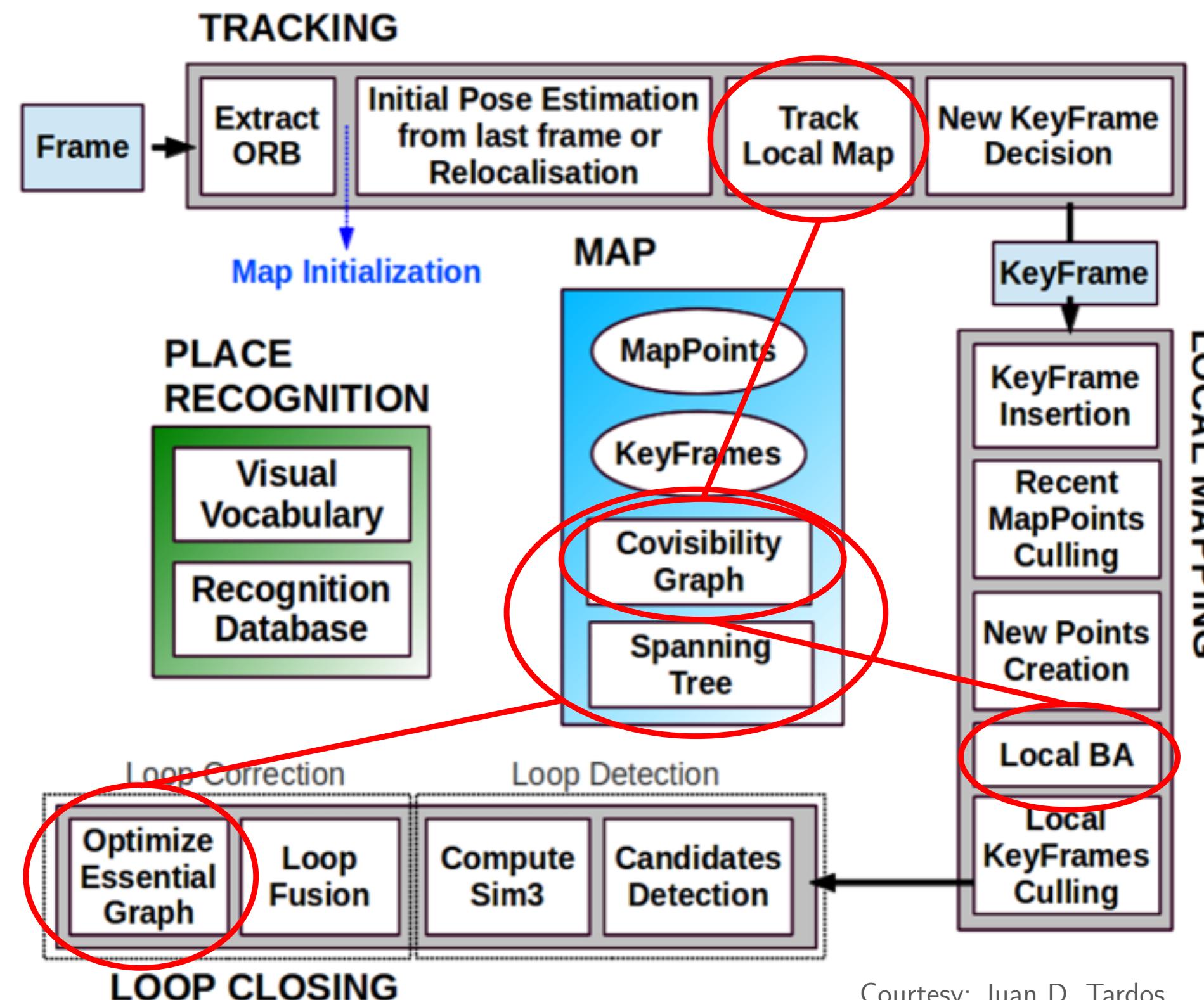
1. Map Module
2. Place Recognition Module
3. Tracking Thread
4. Local Mapping Thread
5. Loop Closing Thread

# Map Module: Key Concepts



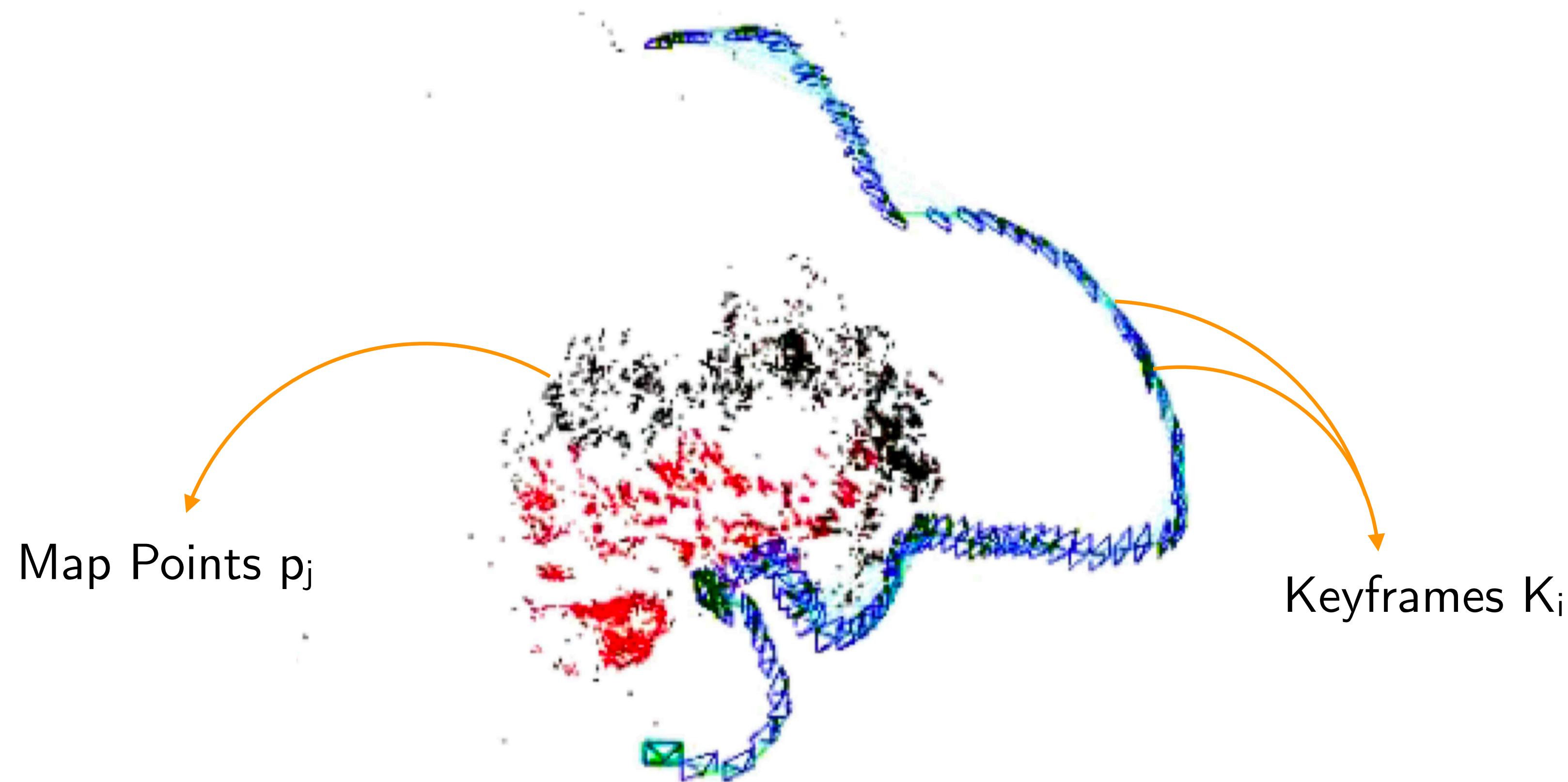
# Map Module: Where is it used?

- By tracking thread for tracking a local map
- By mapping thread to do local bundle adjustment
- By loop closing thread to optimize essential graph

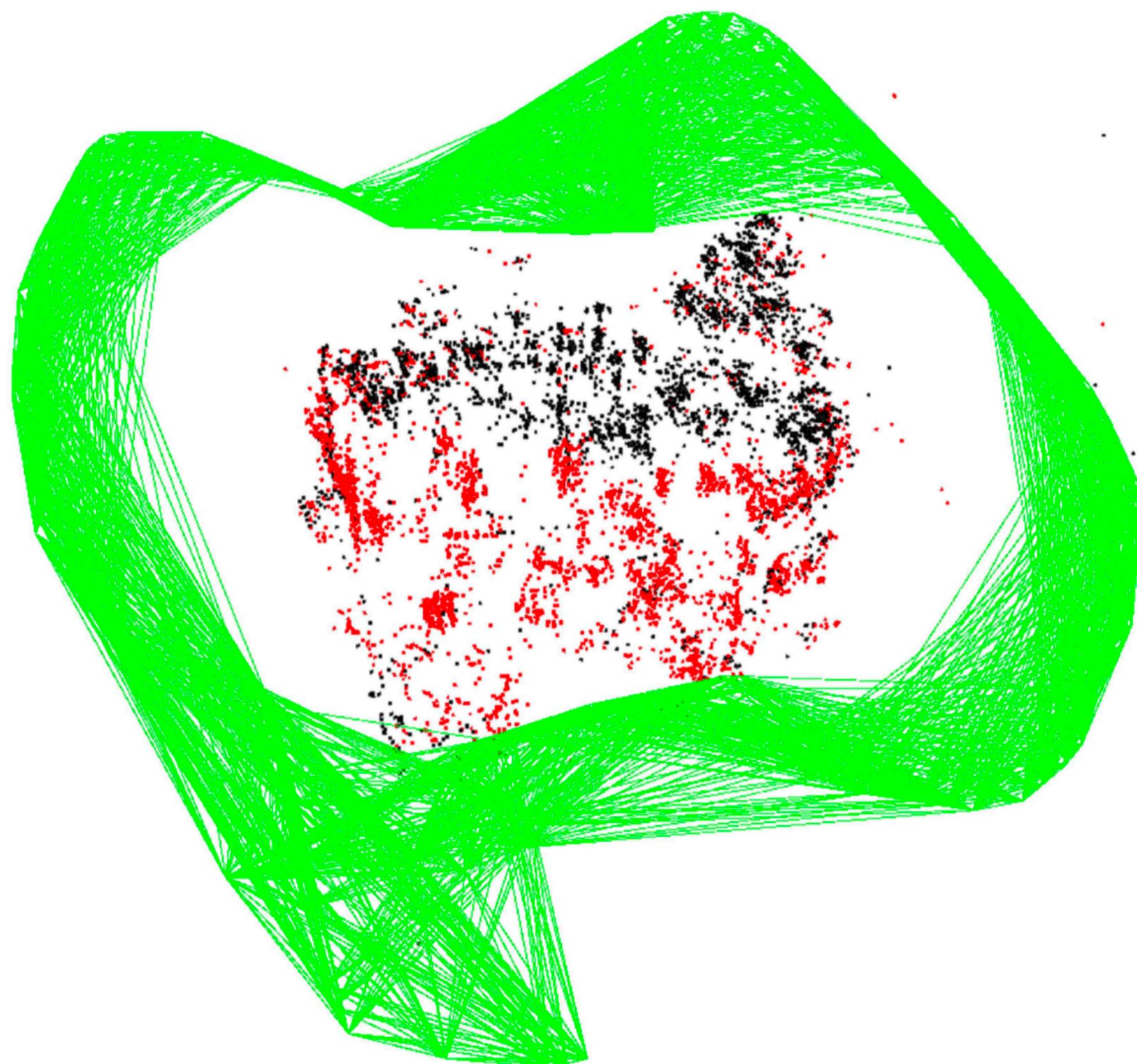


Courtesy: Juan D. Tardos

# Map points, Keyframes



# Covisibility Graph



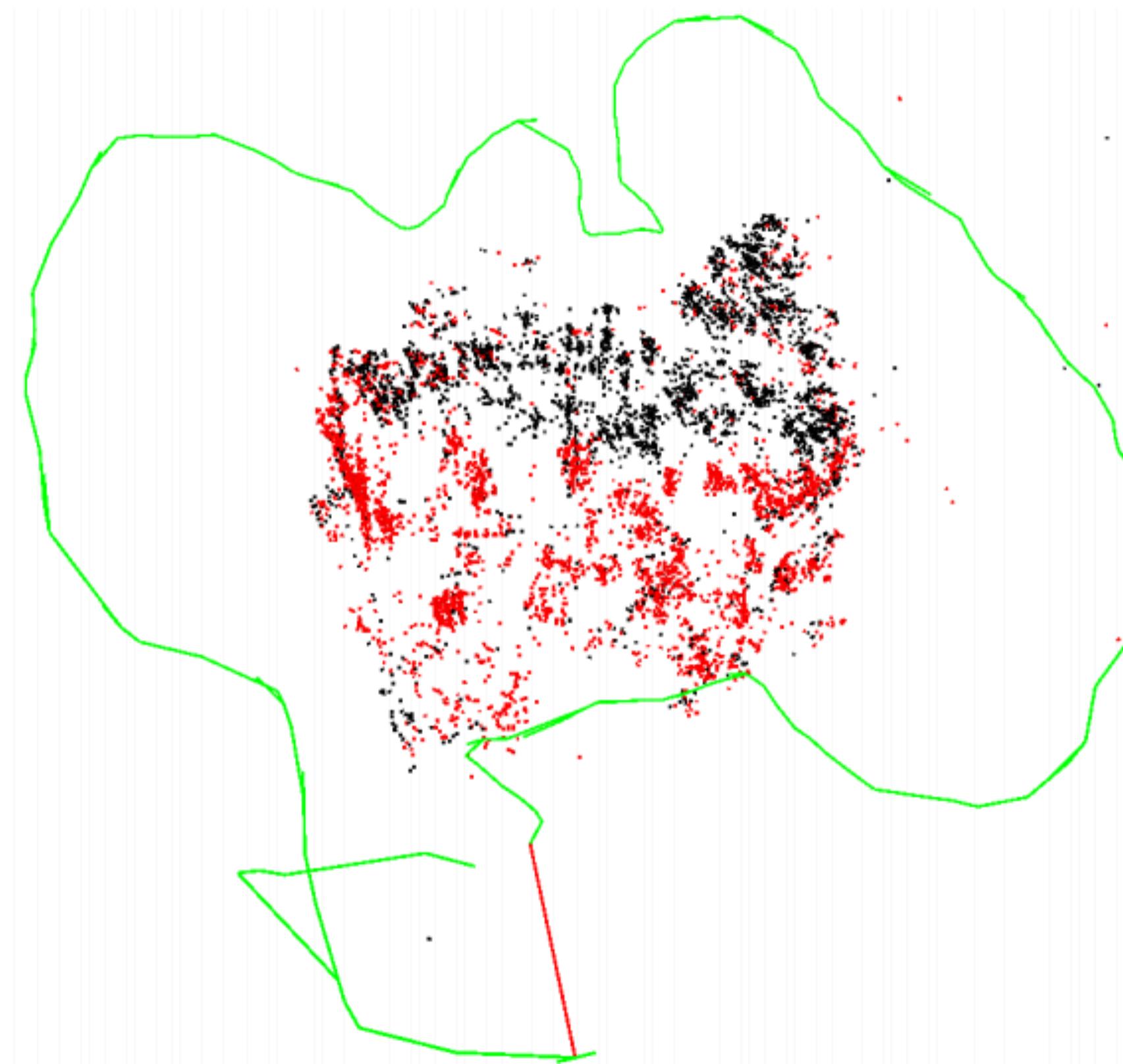
Undirected weighted graph

Each node a keyframe

Each edge weight the number of  
common map points

Edge exists if the two keyframes  
share  $\geq 15$  same map points

# Spanning Tree

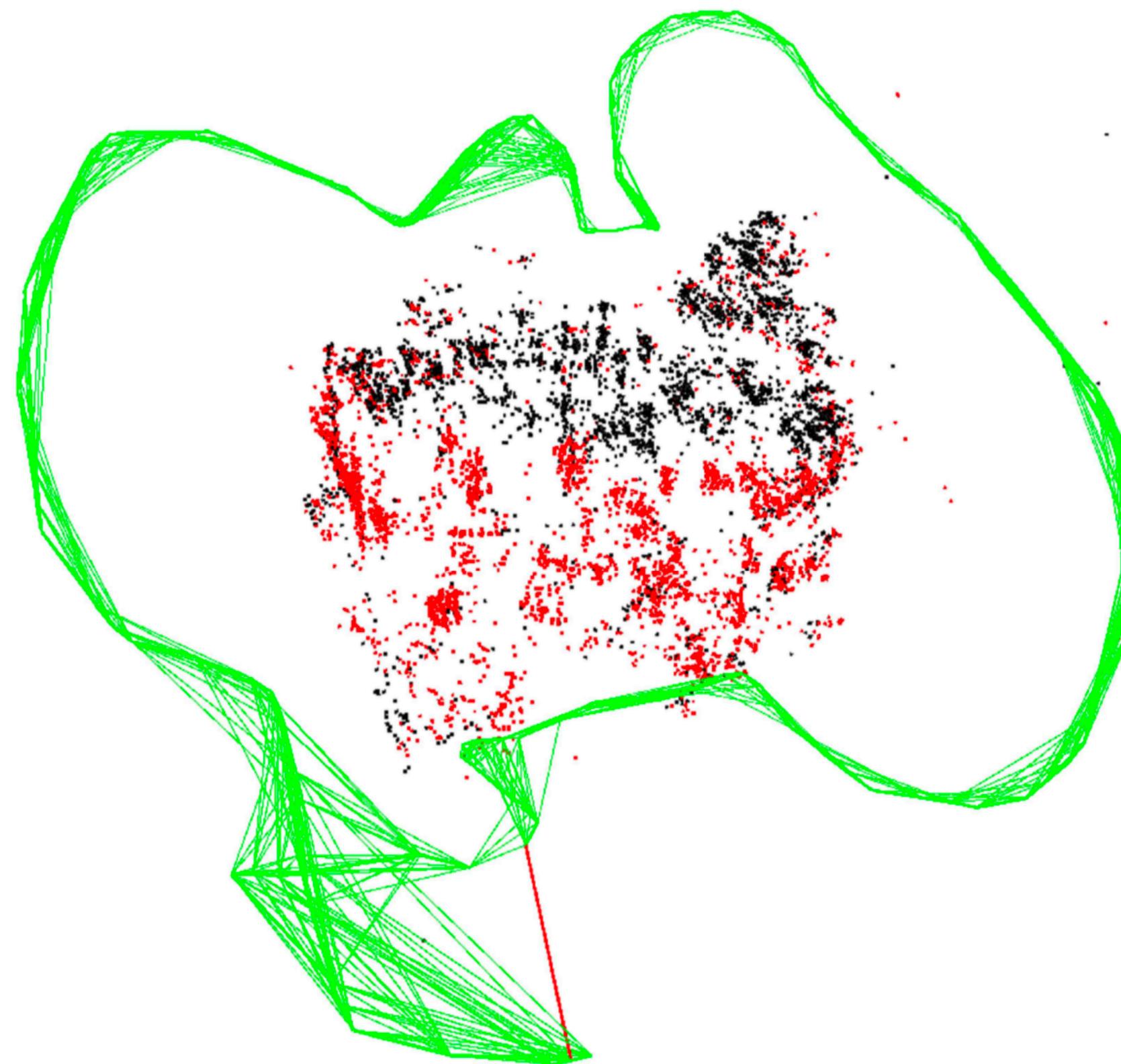


Minimal spanning tree with all vertices from covisibility graph but minimal number of edges

Connected subgraph of covisibility graph

Built incrementally

# Essential Graph

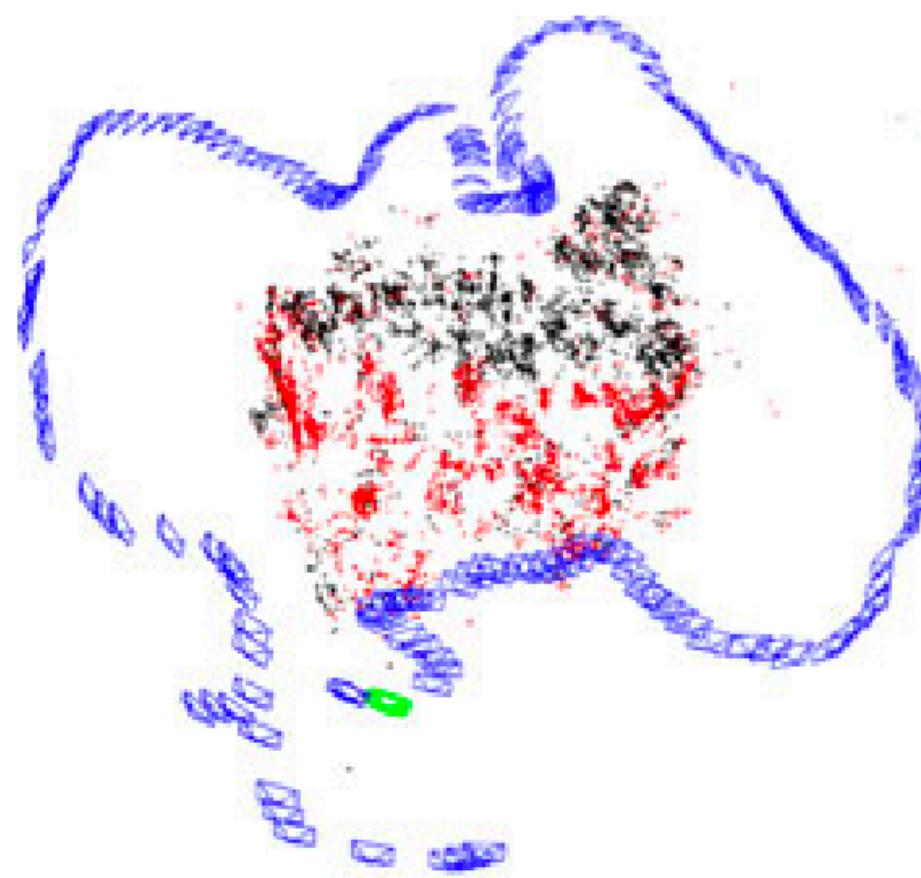


Spanning tree + subset of edges from covisibility graph

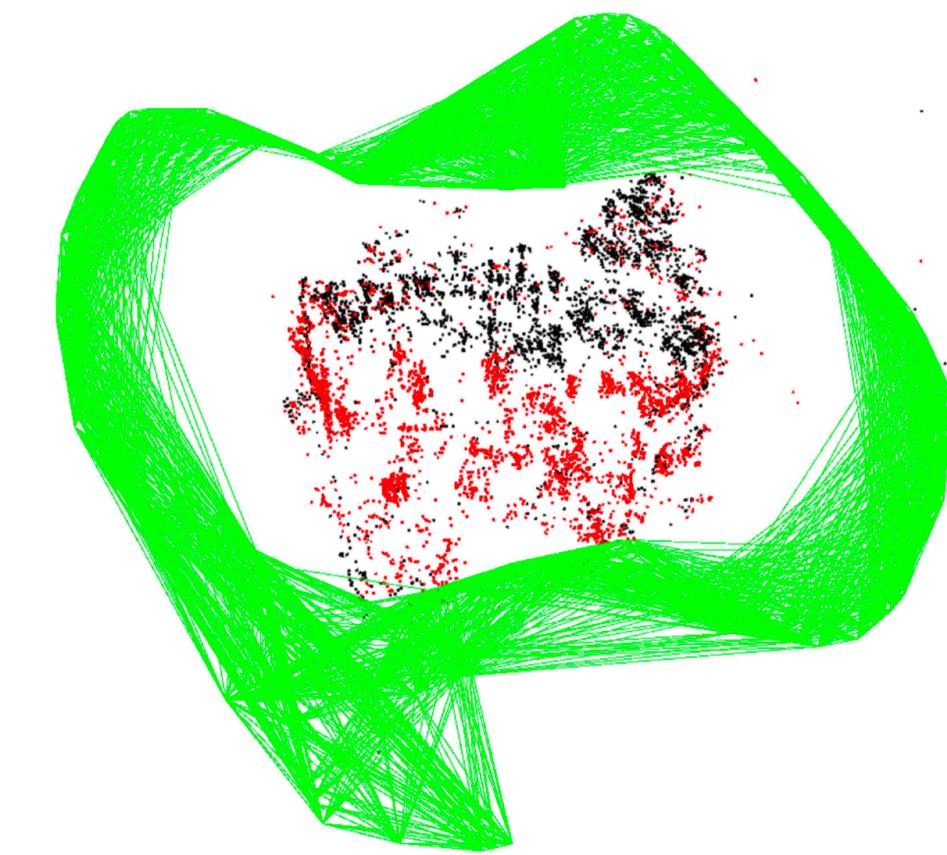
Subset of edges chosen: High covisibility edges ( $\text{weights} \geq 100$ ) + loop closure edges

Stronger network of cameras over the spanning tree but not as dense as covisibility graph

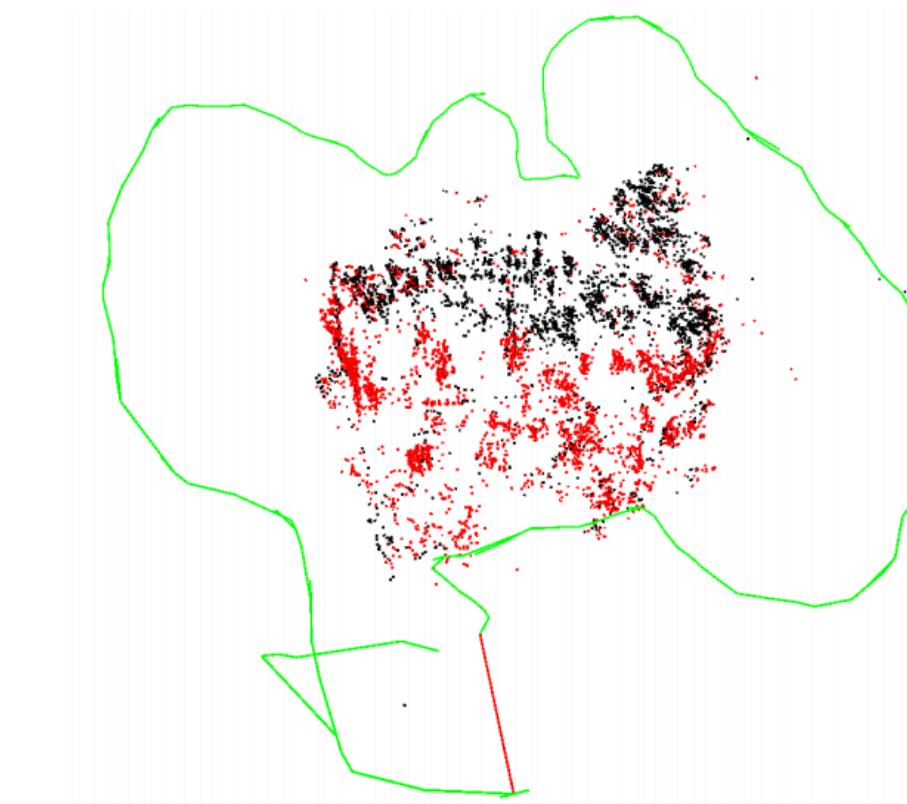
Map of Keyframes  
and 3D points



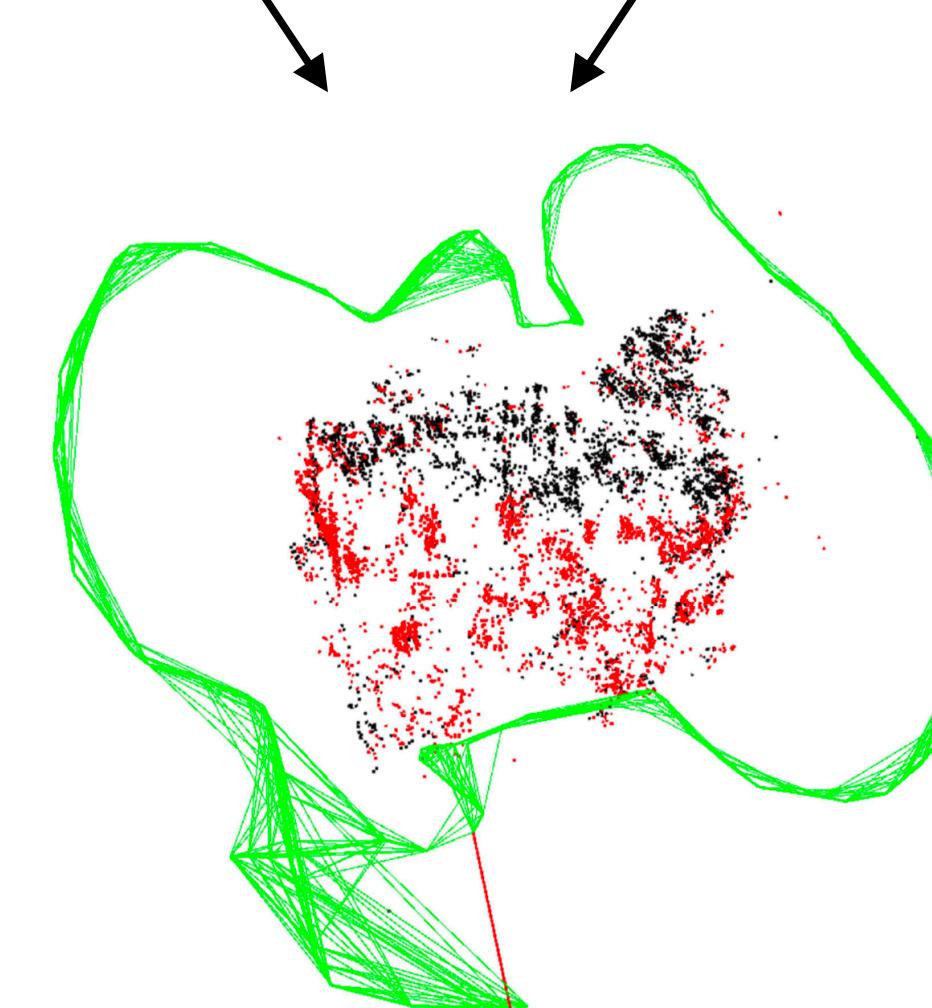
Covisibility Graph



Spanning Tree



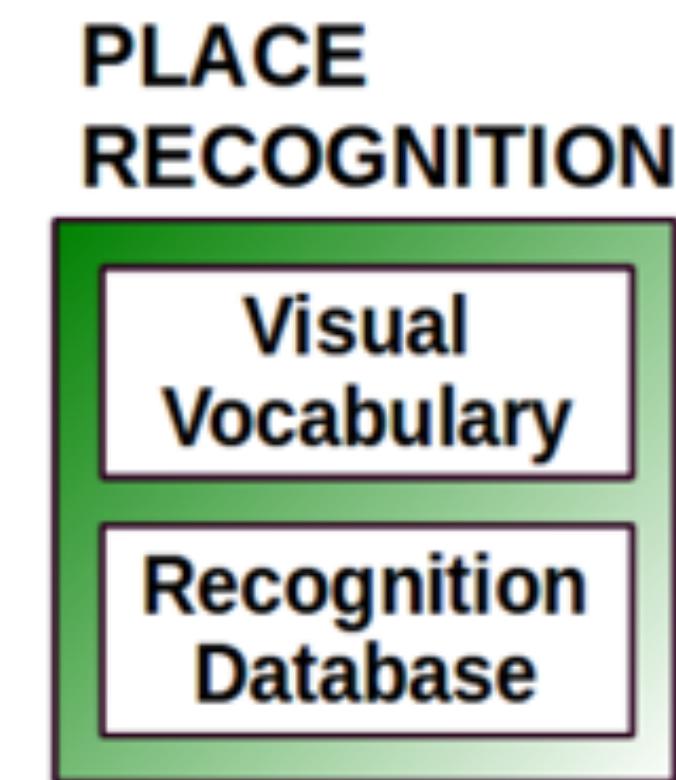
Essential Graph



# Outline

1. Map Module
2. Place Recognition Module
3. Tracking Thread
4. Local Mapping Thread
5. Loop Closing Thread

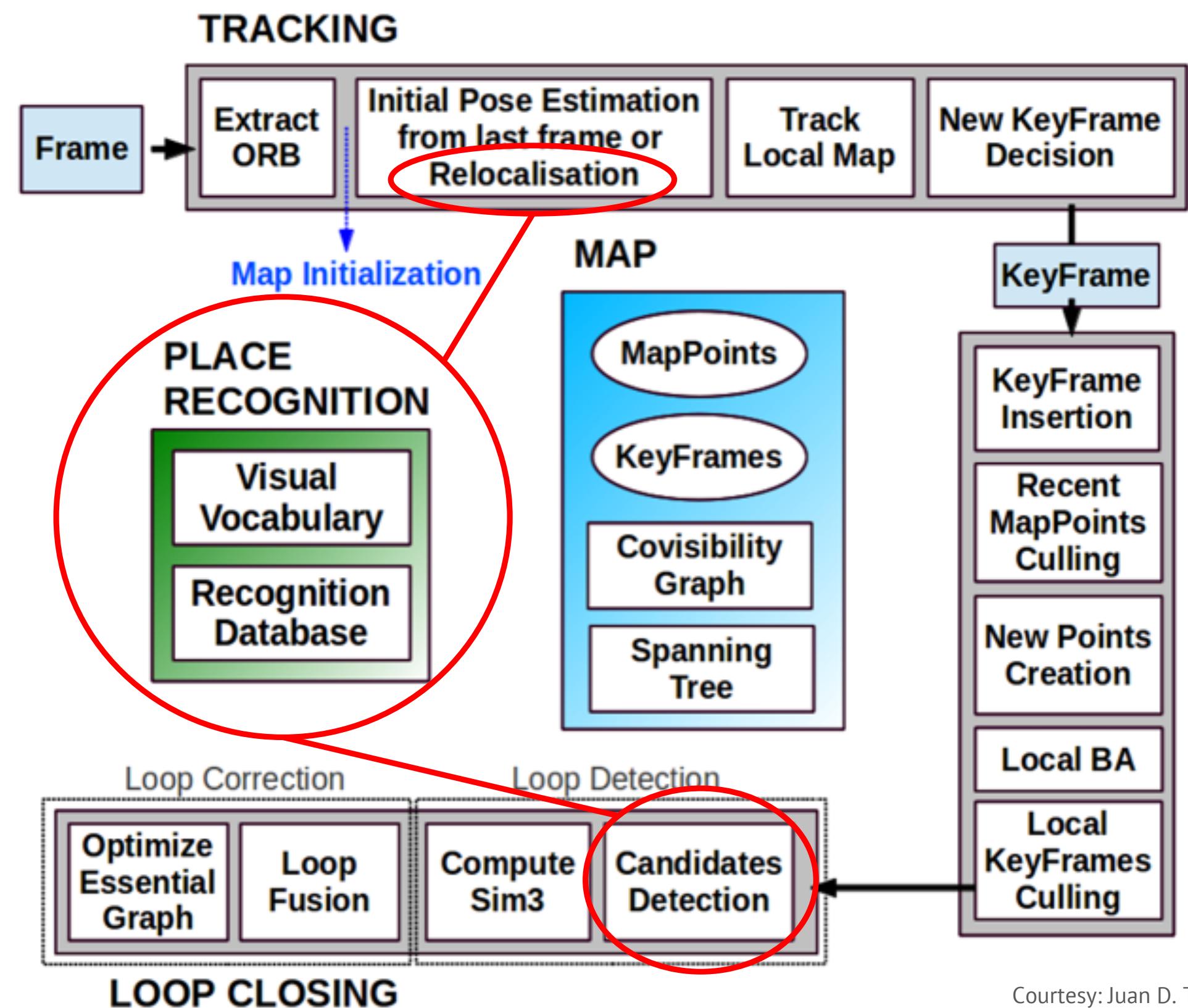
# Place Recognition Module: Key Concepts



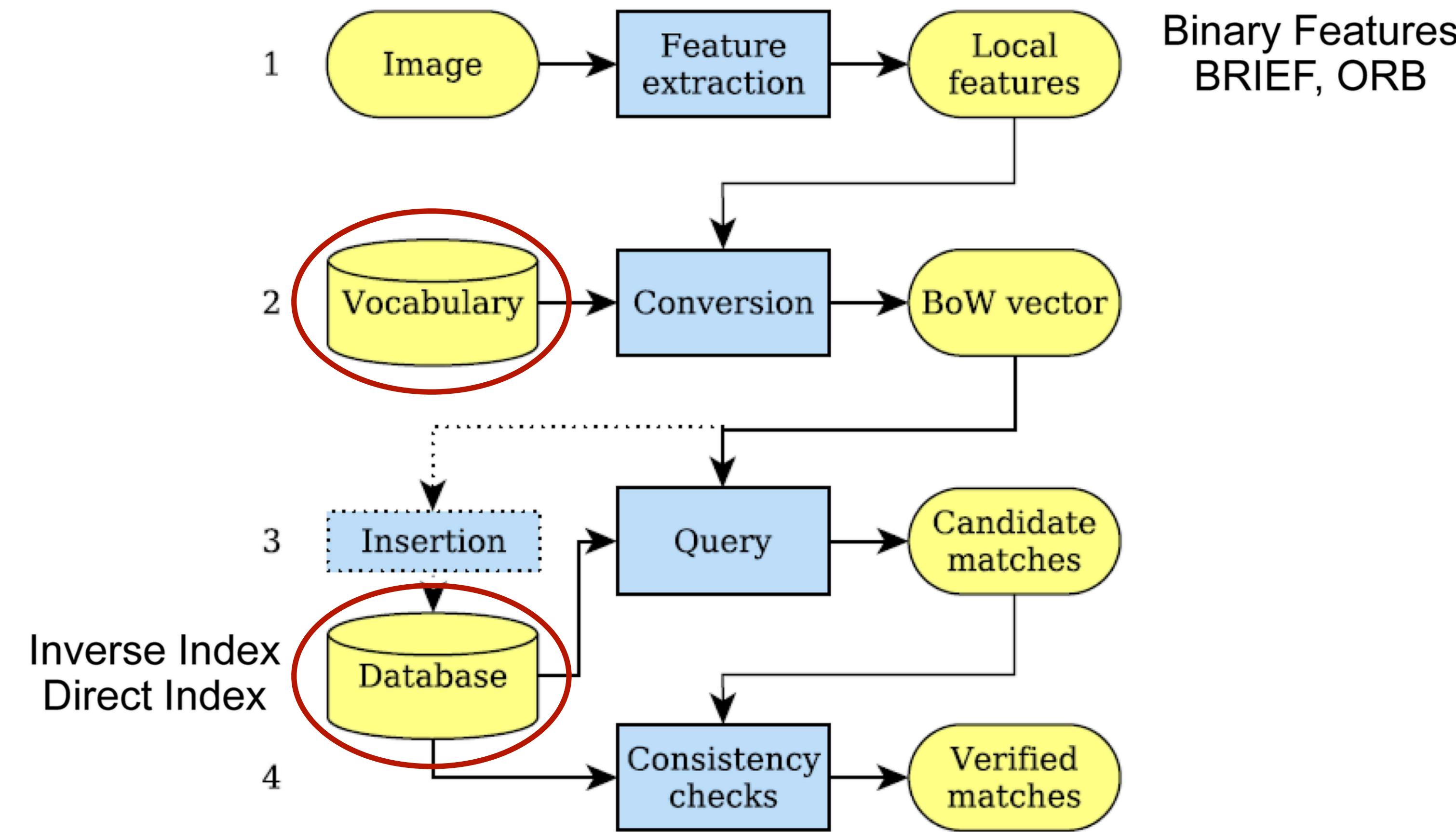
# Place Recognition Module: Where is it used?

By tracking thread for **global relocation** under **tracking failure**

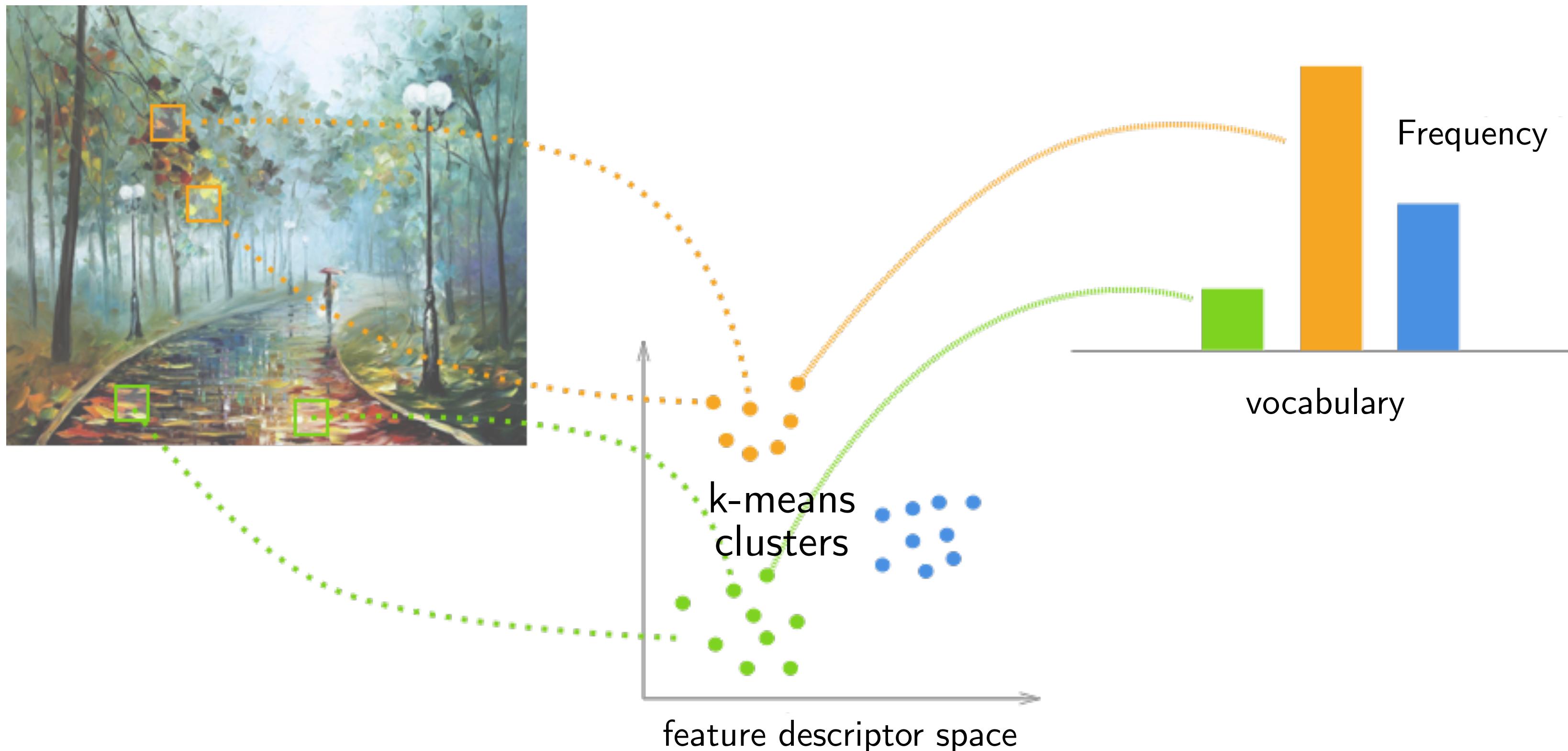
By loop closing thread for loop candidates detection



# Recognition Database



# Vocabulary: Visual Vocabulary Histogram



Basic visual bag-of-words with **fixed number of clusters k-means**

# Vocabulary: Visual Vocabulary Tree

Hierarchical bag-of-words library (**DBoW2**) [1] used in ORB-SLAM

DBoW2 builds a vocabulary tree discretizing a binary FAST+BRIEF feature descriptor space

Vocabulary tree [2] constructed using **hierarchical k-means** clustering

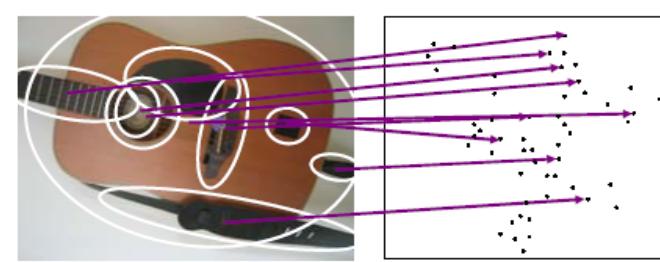
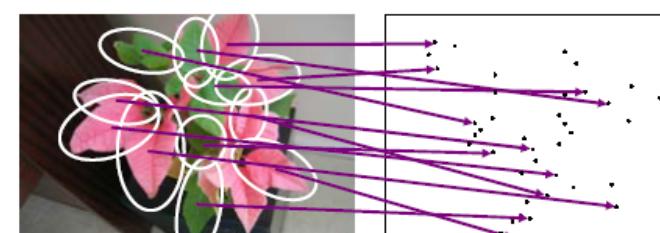
**Tree structure** allows larger and more discriminatory vocabulary to be used **efficiently**

[1] Gálvez-López, Dorian, and Juan D. Tardos. "Bags of binary words for fast place recognition in image sequences." *IEEE Transactions on Robotics* 28.5, 2012: 1188-1197.

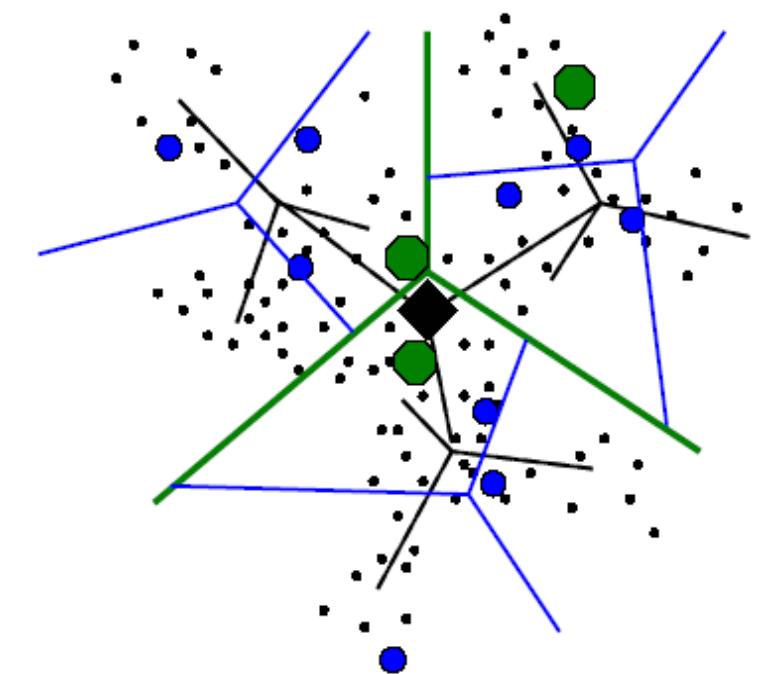
[2] Nister, David, and Henrik Stewenius. "Scalable recognition with a vocabulary tree." *IEEE computer society conference on Computer vision and pattern recognition*, 2006.

# Vocabulary: Visual Vocabulary Tree

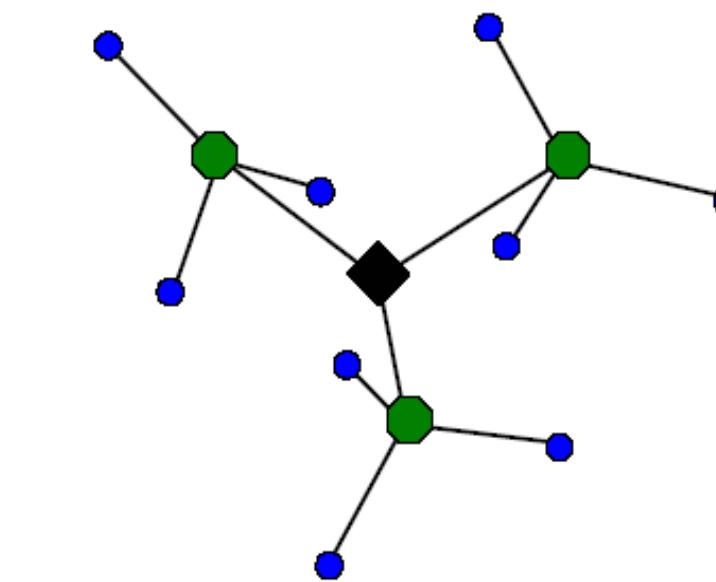
## Train Vocabulary (Offline)



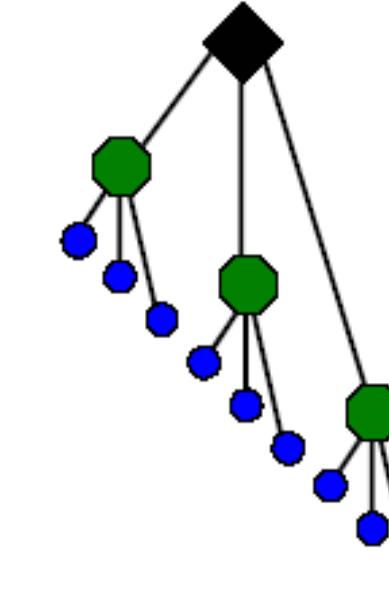
Extract Features



Hierarchical k-means



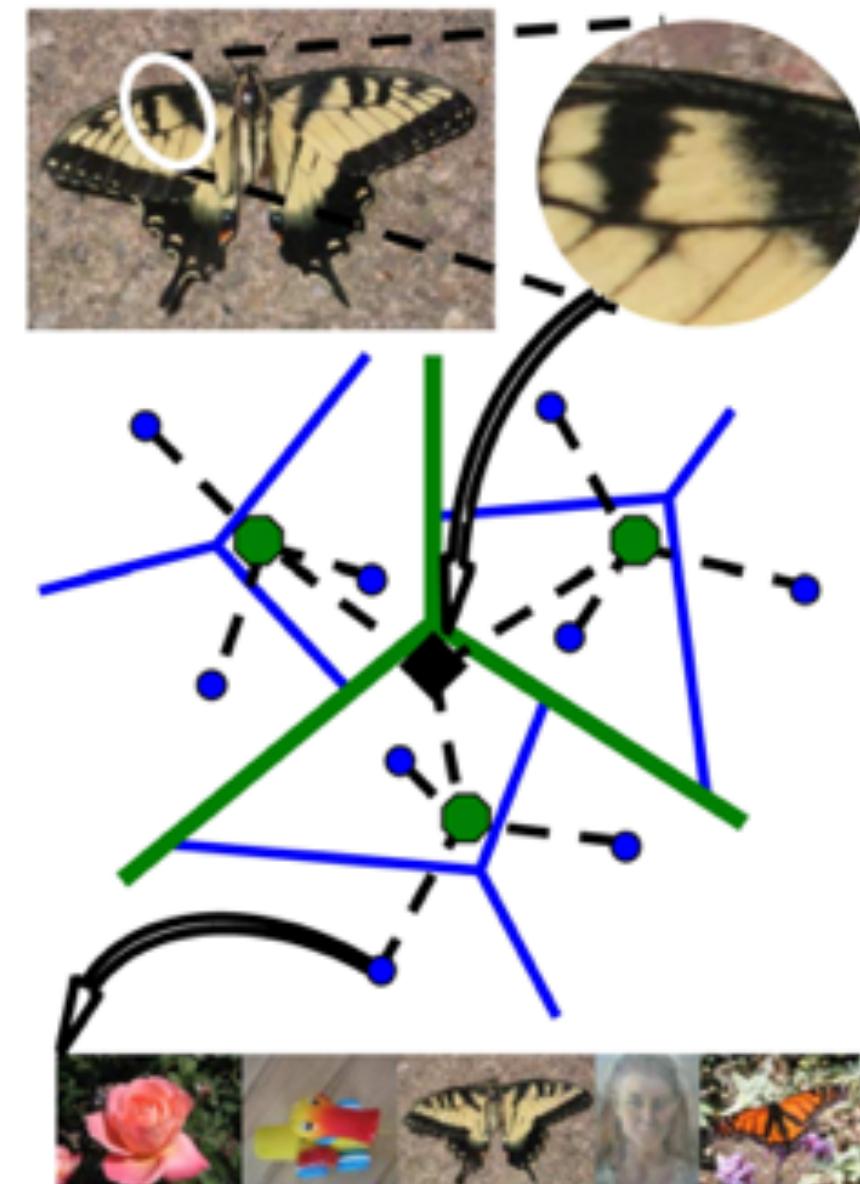
k-means centers



Vocabulary Tree

# Vocabulary: Visual Vocabulary Tree

Insertion/Query Database (Online)



vocabulary tree with branch factor  
three, quantization levels two

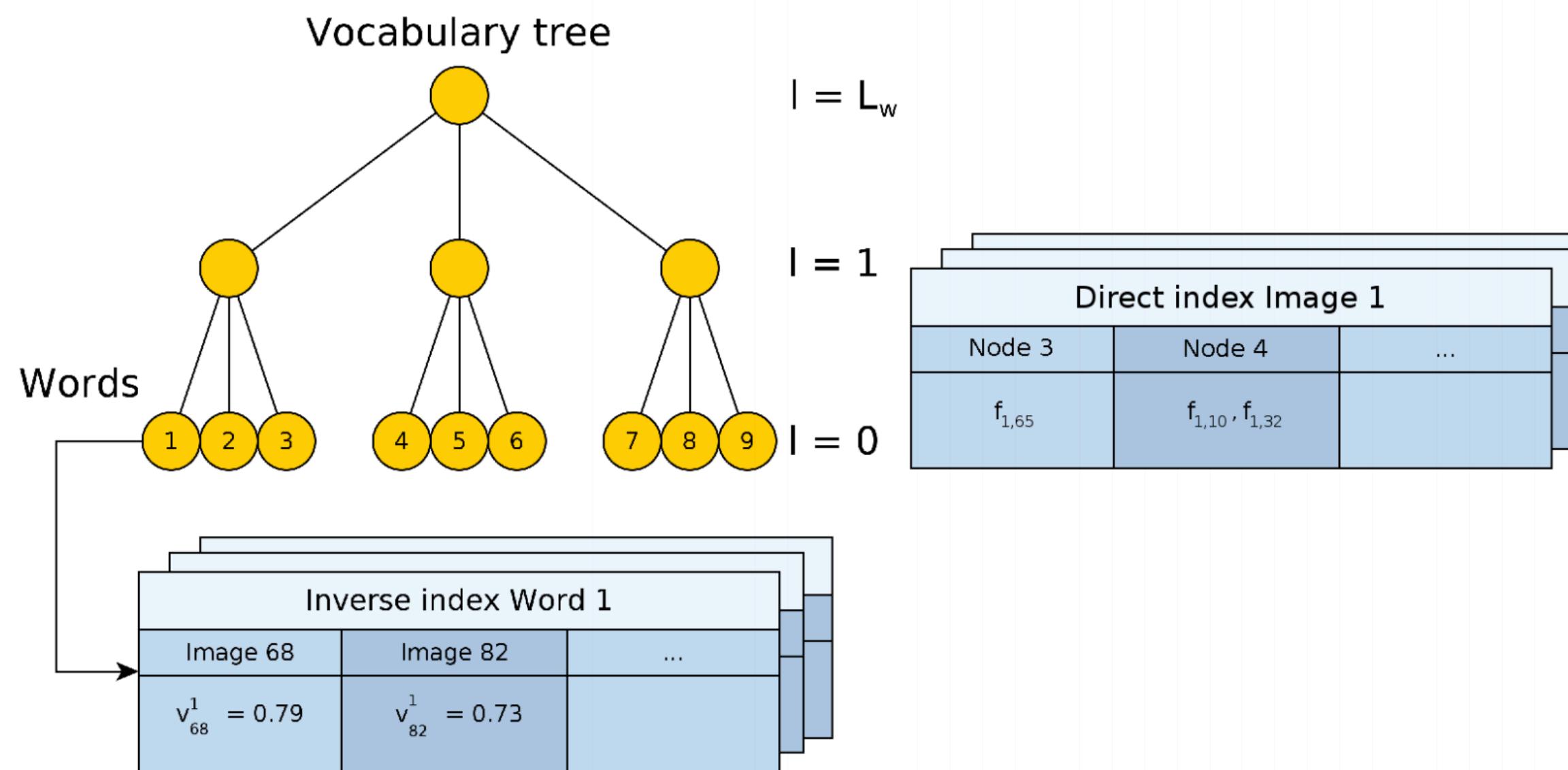
Hierarchical quantization levels

At **first** quantization level:  
Descriptor assigned to closest of 3  
**green** centers

At **second** quantization level:  
Descriptor assigned to closest of 3  
**blue** centers

# Database: Direct, Inverse Indexes

DBoW2 vocabulary tree consists of an **inverse** and **direct** index



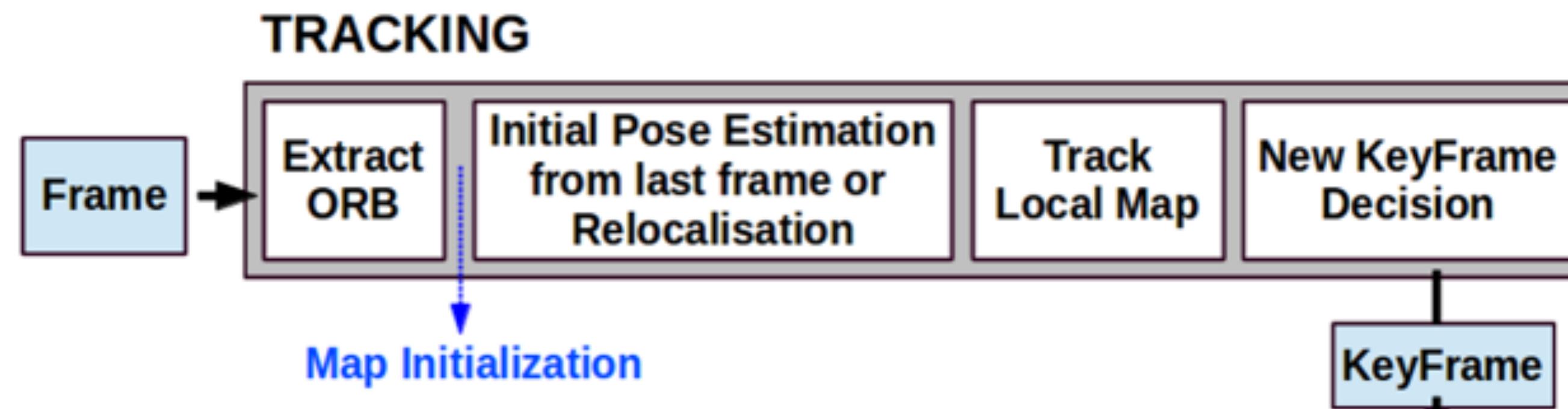
**Inverse Index:** Used for fast retrieval of images potentially similar to a given one

**Direct Index:** Used to efficiently obtain point correspondences between images

# Outline

1. Map Module
2. Place Recognition Module
3. Tracking Thread
4. Local Mapping Thread
5. Loop Closing Thread

# Tracking: Key Concepts



# Automatic Map Initialization

Automatic Model Selection between,

Fundamental/Essential Matrix (General Scenes)

Homography (Planar or Low Parallax Scenes)

Problem	Inputs	Model to find	Basic Equation	d.o.f.	Min. # of matches	Minimal solution
Initialize 3D scene	$\mathbf{u}_{1j}, \mathbf{u}_{2j}$	Essential Matrix $\mathbf{E}_{12} = [\mathbf{t}]_{\times} \mathbf{R}$	$\mathbf{u}_{1j}^T \mathbf{E}_{12} \mathbf{u}_{2j} = 0$	5	5	5-point 8-point
Initialize 2D scene	$\mathbf{u}_{1j}, \mathbf{u}_{2j}$	Homography $\mathbf{H}_{12}$	$\mathbf{u}_{1j} = \mathbf{H}_{12} \mathbf{u}_{2j}$	8	4	

# Key Ideas for Improved Tracking

ORB: FAST corner + Oriented Rotated Brief descriptor

- **Binary** descriptor
- Very **fast** to compute and compare

Use same features for Tracking, Mapping, Loop Closing, Relocalization

“**Survival of fittest**” strategy for points and keyframes: add to points and keyframes to map at a high rate and cull redundant points and keyframes

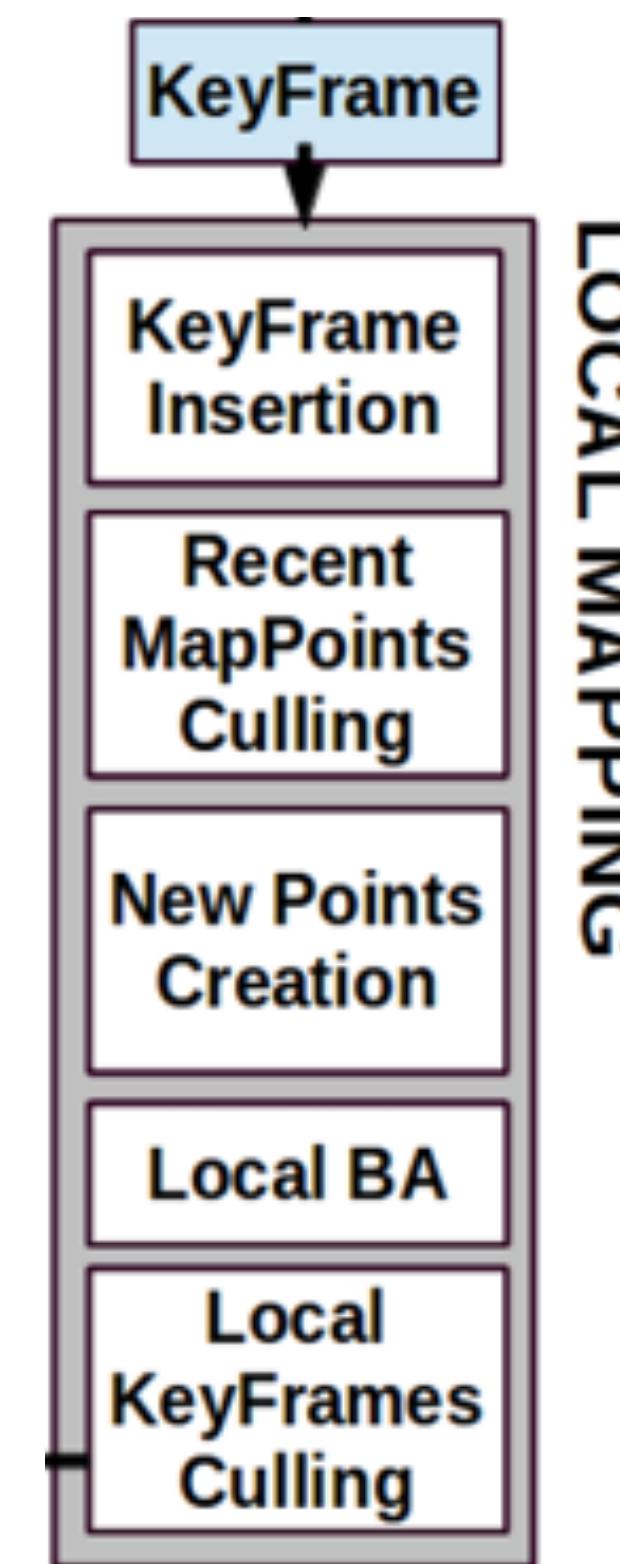
Only **tracks a local map**, computed using the covisibility graph

If tracking lost, **global relocalization** via place recognition module

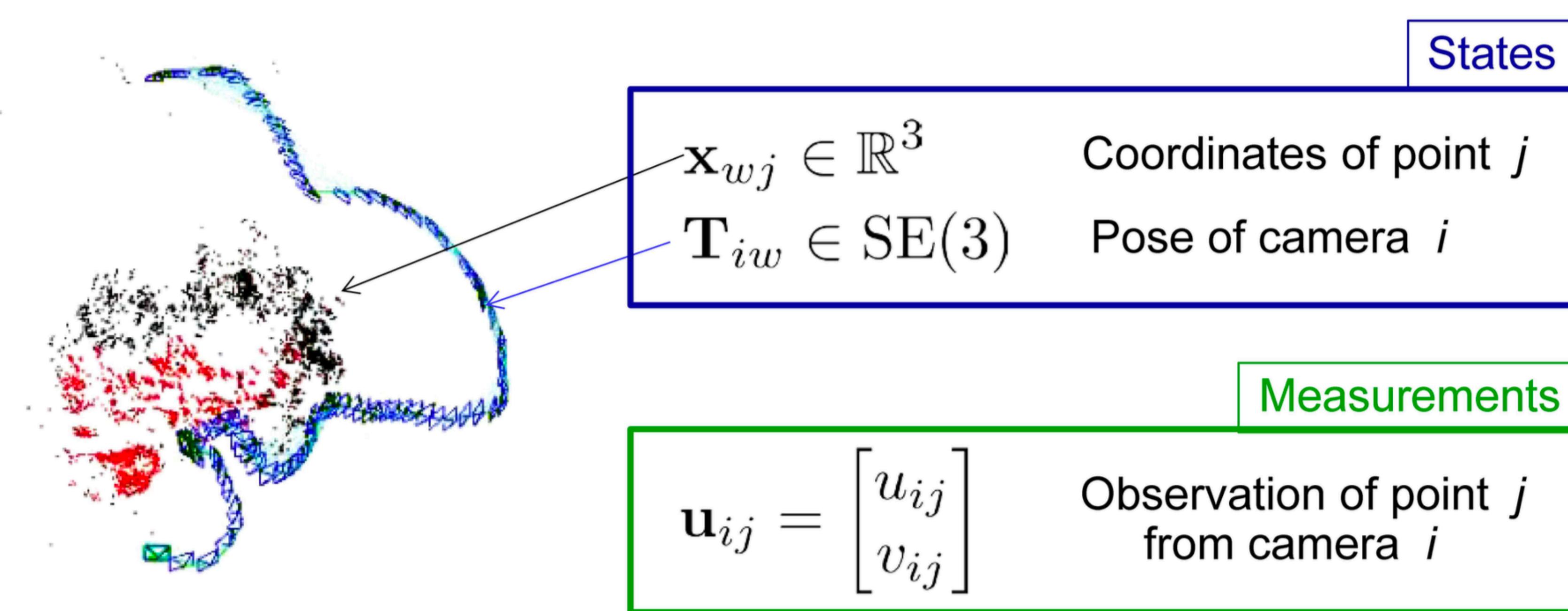
# Outline

1. Map Module
2. Place Recognition Module
3. Tracking Thread
4. Local Mapping Thread
5. Loop Closing Thread

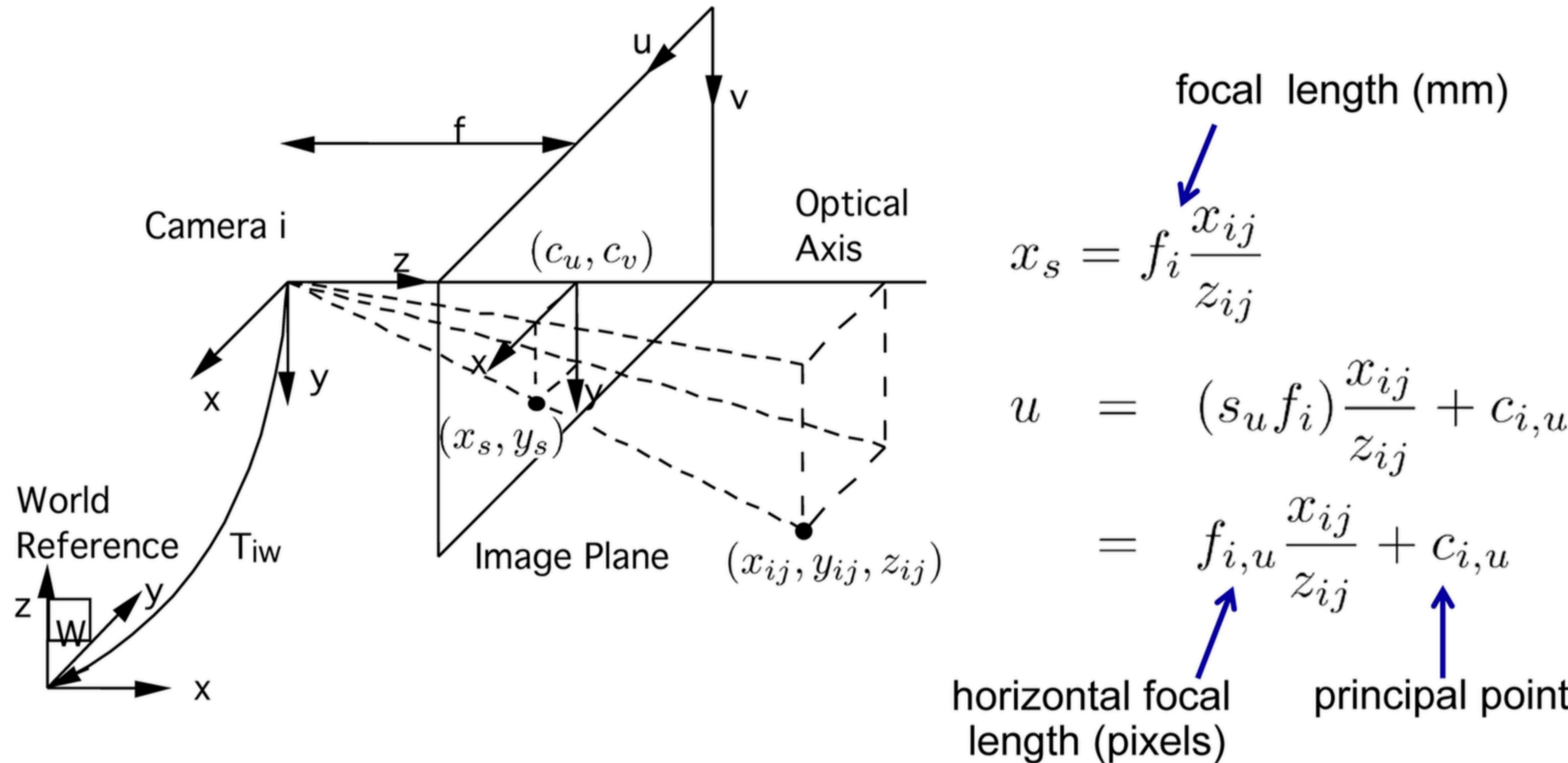
# Local Mapping: Key Concepts



# Local Bundle Adjustment (BA)



# Local BA: Camera Projection Model



- In summary:

$$\pi_i(\mathbf{T}_{iw}, \mathbf{x}_{wj}) = \begin{bmatrix} f_{i,u} \frac{x_{ij}}{z_{ij}} + c_{i,u} \\ f_{i,v} \frac{y_{ij}}{z_{ij}} + c_{i,v} \end{bmatrix}$$

# Local BA: Optimization

Optimizes a **local map** that is potentially visible,

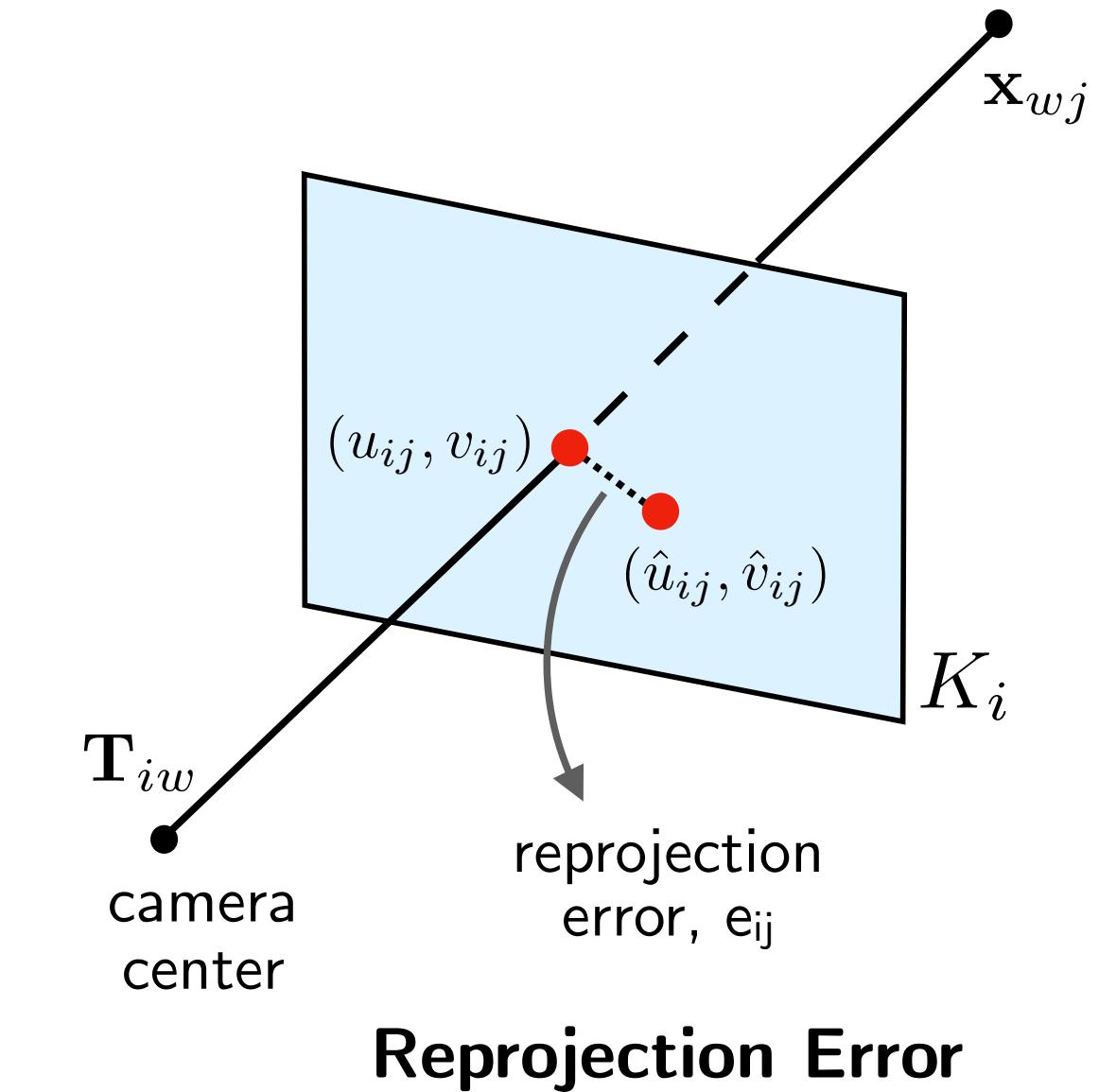
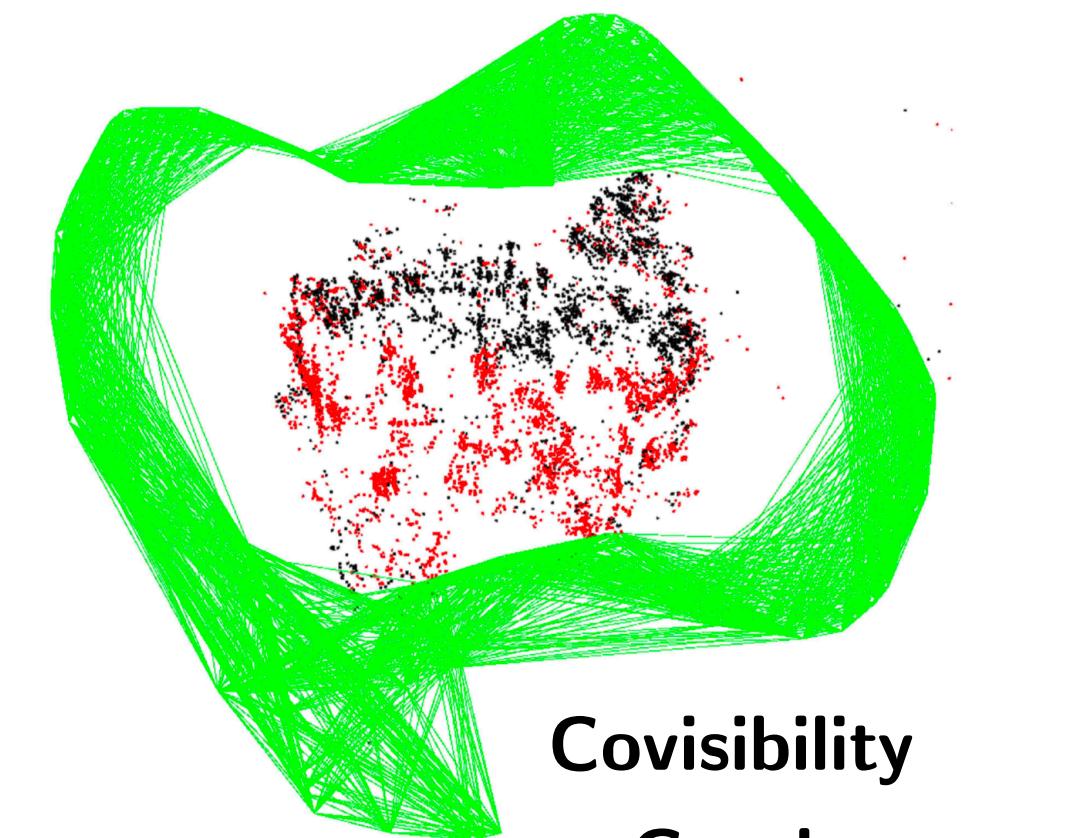
Keyframes  $i$  include current keyframe and all  
keyframe connected to it in the covisibility graph

Map Points  $j$  seen by keyframes  $i$

$$\mathbf{e}_{ij} = \mathbf{u}_{ij} - \pi_i(\mathbf{T}_{iw}, \mathbf{x}_{wj})$$

Bundle Adjustment

$$\{\mathbf{T}_{1w}.. \mathbf{T}_{nw}, \mathbf{x}_{w1}.. \mathbf{x}_{wm}\}^* = \arg \min_{\mathbf{T}, \mathbf{x}} \sum_{i,j} \rho_h(\mathbf{e}_{ij}^T \Sigma_{ij}^{-1} \mathbf{e}_{ij})$$



# Local BA: Robust Estimators

- L2 cost (quadratic)

$$J_{L2}(\theta) = \frac{1}{2} \sum_{i=1}^N \left( h_\theta(\mathbf{x}^{(i)}) - y^{(i)} \right)^2$$

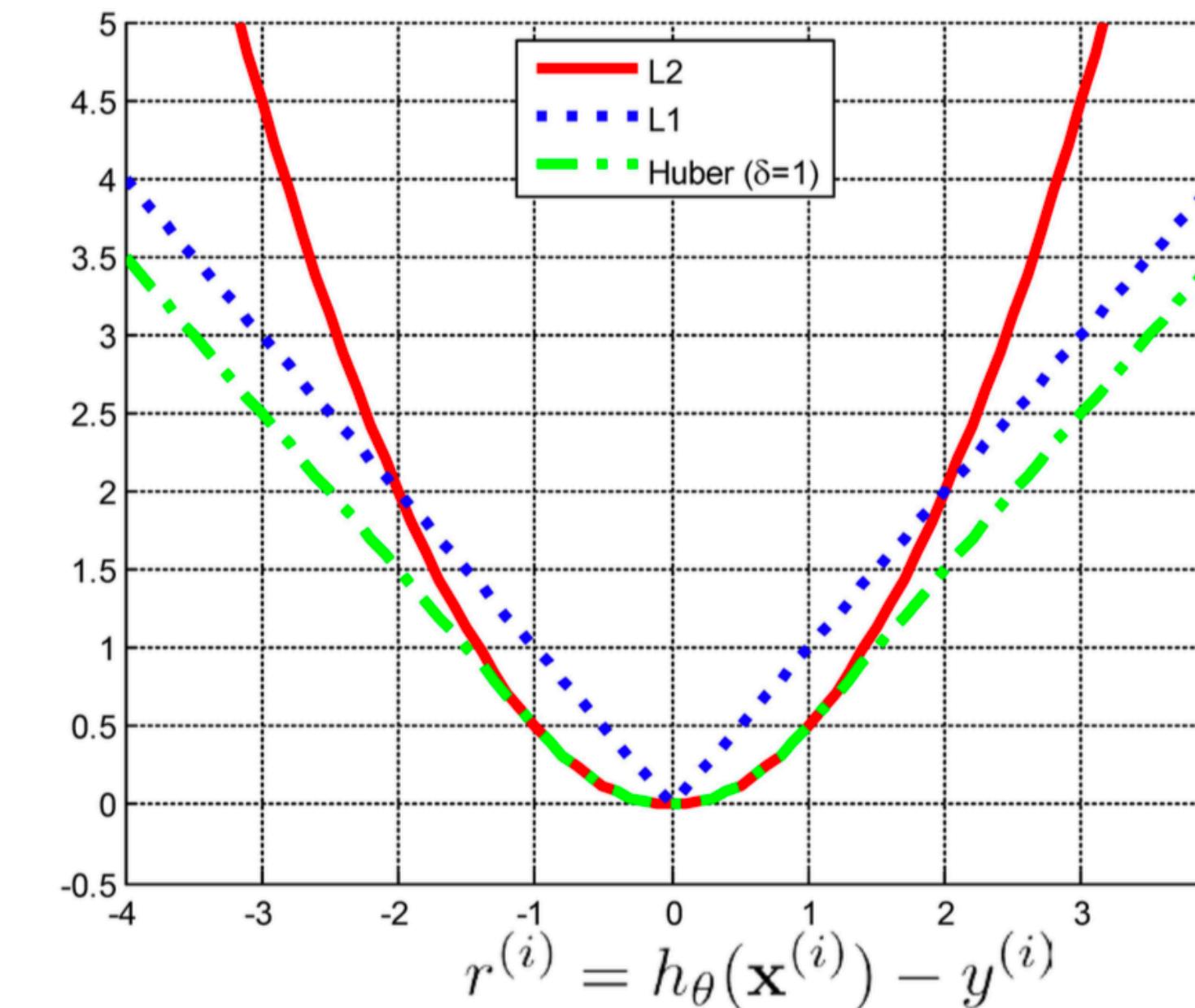
- L1 cost (absolute value)

$$J_{L1}(\theta) = \sum_{i=1}^N \left| h_\theta(\mathbf{x}^{(i)}) - y^{(i)} \right|$$

- Huber cost:

$$L_H(r, \delta) = \begin{cases} r^2/2 & \text{if } |r| \leq \delta \\ \delta|r| - \delta^2/2 & \text{if } |r| > \delta \end{cases}$$

$$J_H(\theta) = \sum_{i=1}^N L_H(r^{(i)}, \delta) = \sum_{|r^{(i)}| \leq \delta} r^{(i)2}/2 + \sum_{|r^{(i)}| > \delta} \delta|r^{(i)}| - \delta^2/2$$

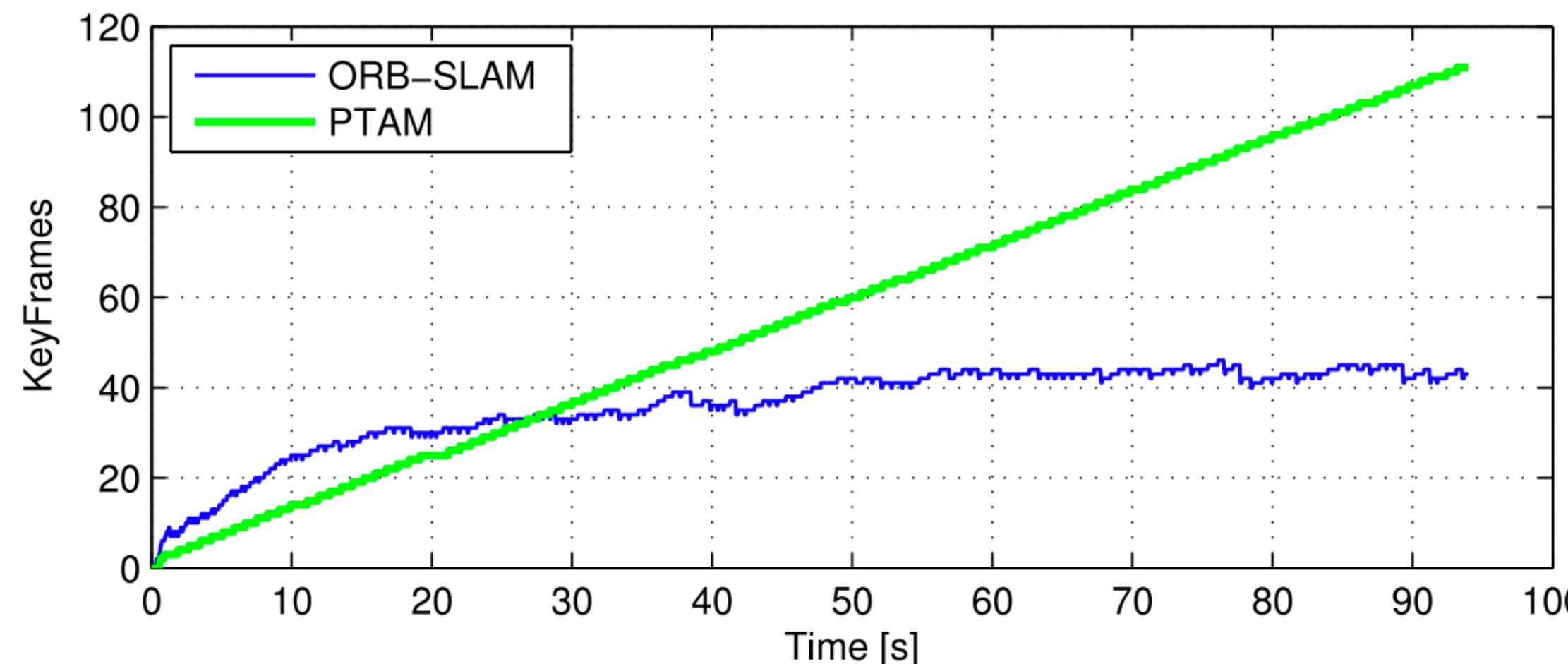


# Local Keyframes Culling

Detect **redundant keyframes** and delete them

Discard all keyframes in essential graph whose 90% map points have been seen in at least three other keyframes in the same/finer scale

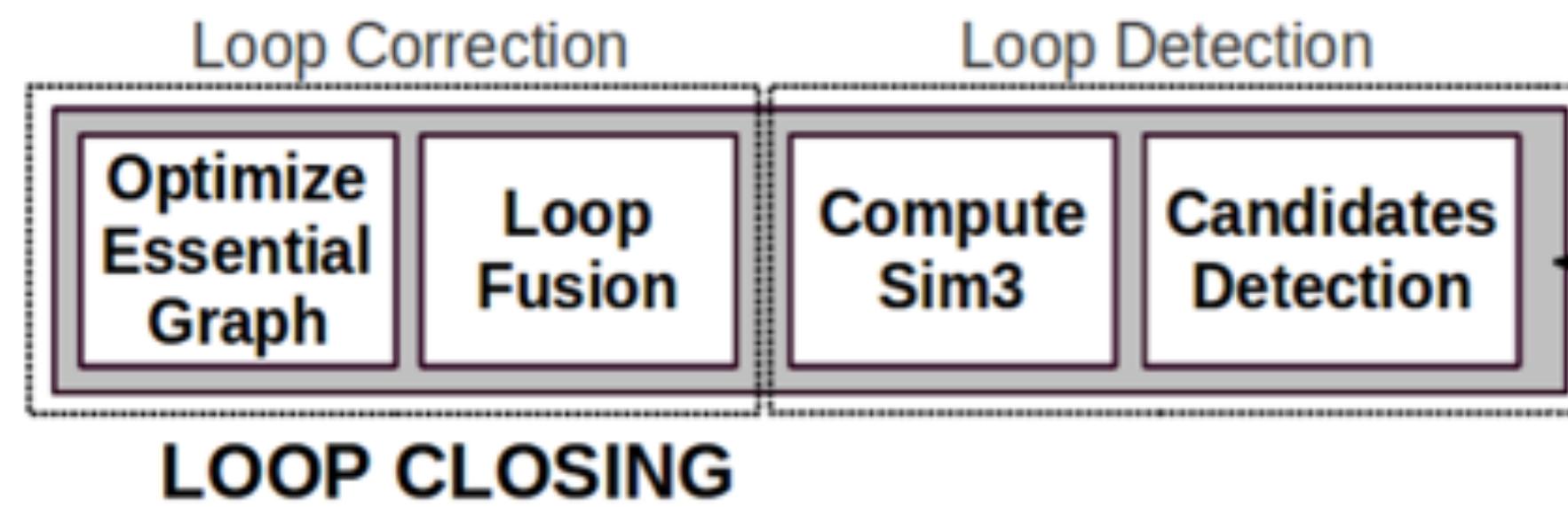
Enables **lifelong operation** in the **same static environment** (no visual changes) as number of keyframes will not grow unbounded



# Outline

1. Map Module
2. Place Recognition Module
3. Tracking Thread
4. Local Mapping Thread
5. Loop Closing Thread

# Loop Closing: Key Concepts



# Loop Detection

For each frame, query a matching image in the Place Recognition Database

**Is this a loop closure?**

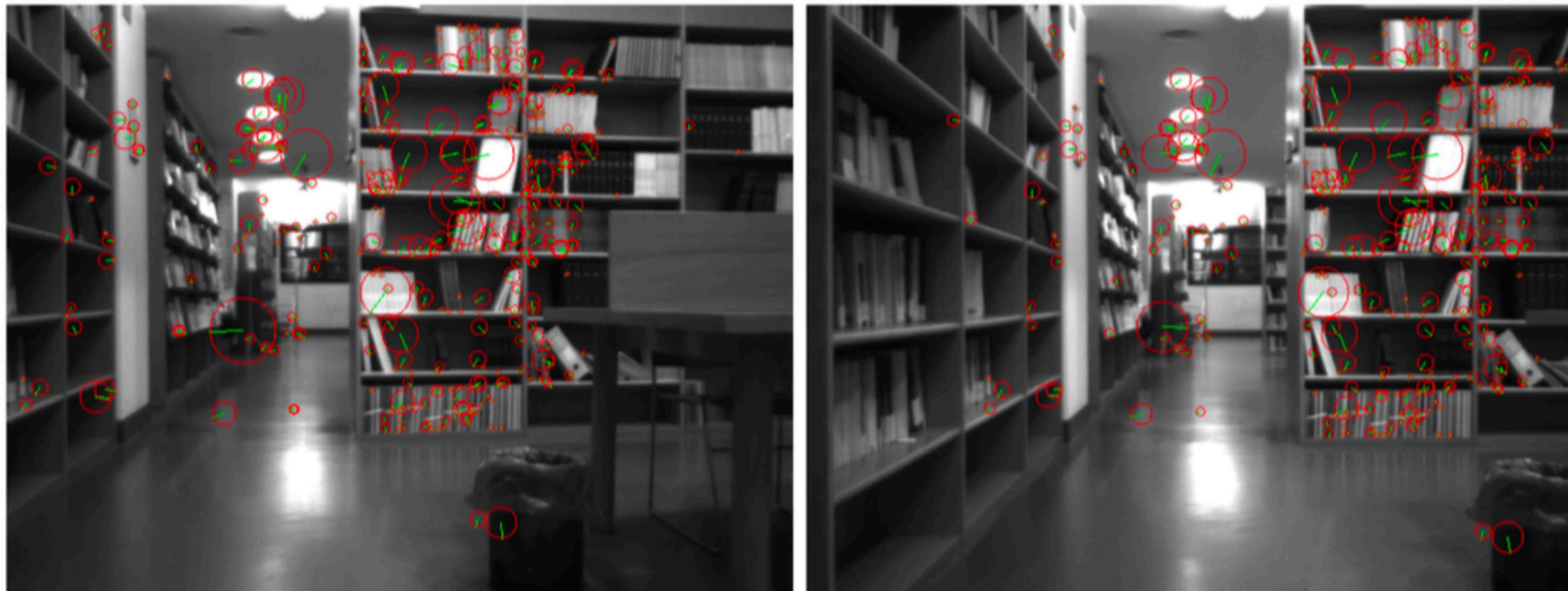


Image Frame Retrieved  
from Database

Current Image Frame

# Loop Detection

For each frame, query a matching image in the Place Recognition Database

**Is this a loop closure?**

Scene 1430



Scene 1244



Image Frame Retrieved  
from Database

Current Image Frame

# Loop Detection

For each frame, query a matching image in the Place Recognition Database

**Is this a loop closure?**

Scene 1430



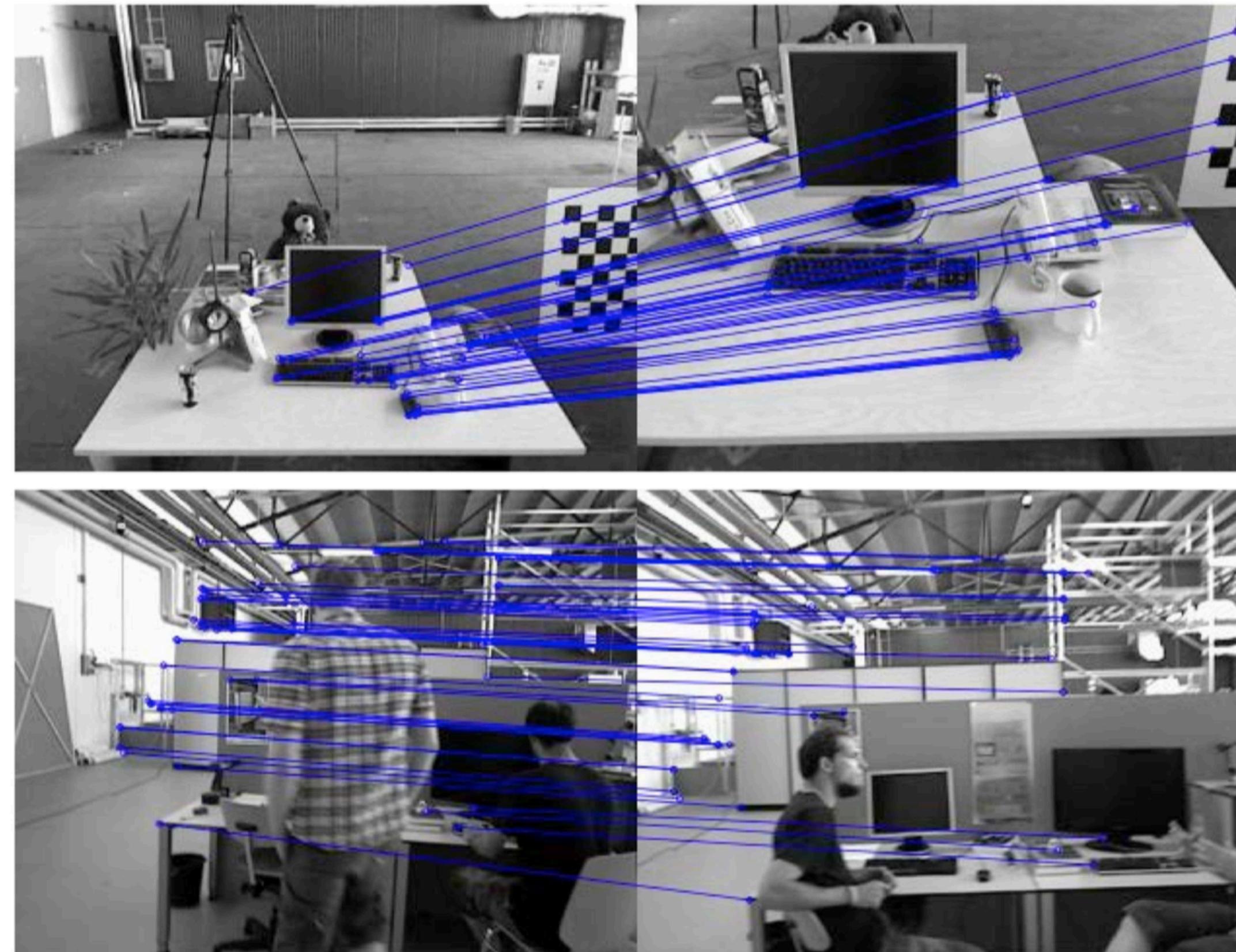
Scene 1244



No, but algorithm can return a false positive! Perceptual Aliasing.

# Loop Detection

Some challenging loop detection examples with DBoW2 using ORB features



# Loop Correction

## Essential Graph Optimization

Pose graph optimization over essential graph to distribute loop closing error along graph

To **correct scale drift**, optimization done over 7-DOF similarity transformations  $\text{Sim}(3)$  instead of 6-DOF rigid body transformations  $\text{SE}(3)$

$$\mathbf{e}_{i,j} = \log_{\text{Sim}(3)}(\mathbf{S}_{ij}\mathbf{S}_{jw}\mathbf{S}_{iw}^{-1})$$

$$C = \sum_{i,j} (\mathbf{e}_{i,j}^T \Lambda_{i,j} \mathbf{e}_{i,j})$$

Map points not included as variables in the optimization (**motion-only BA**)

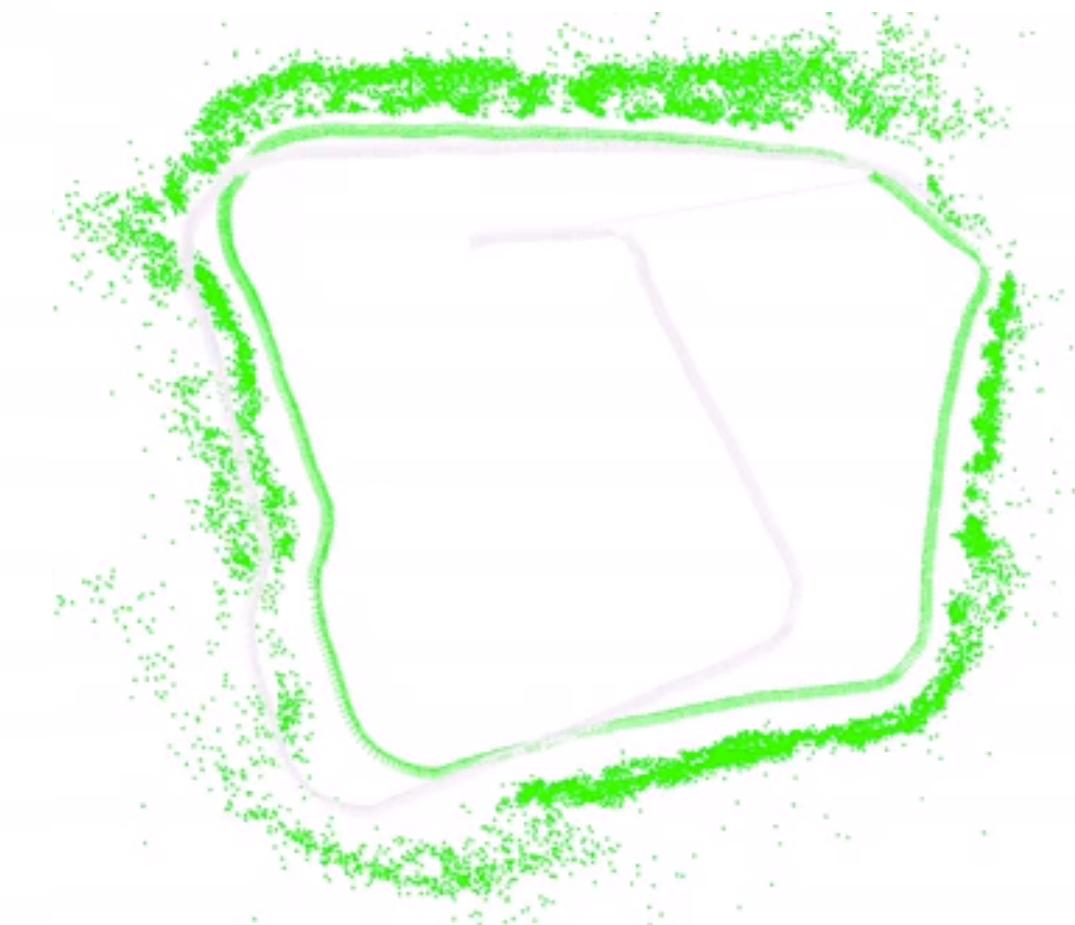
# Loop Correction

Map before optimization



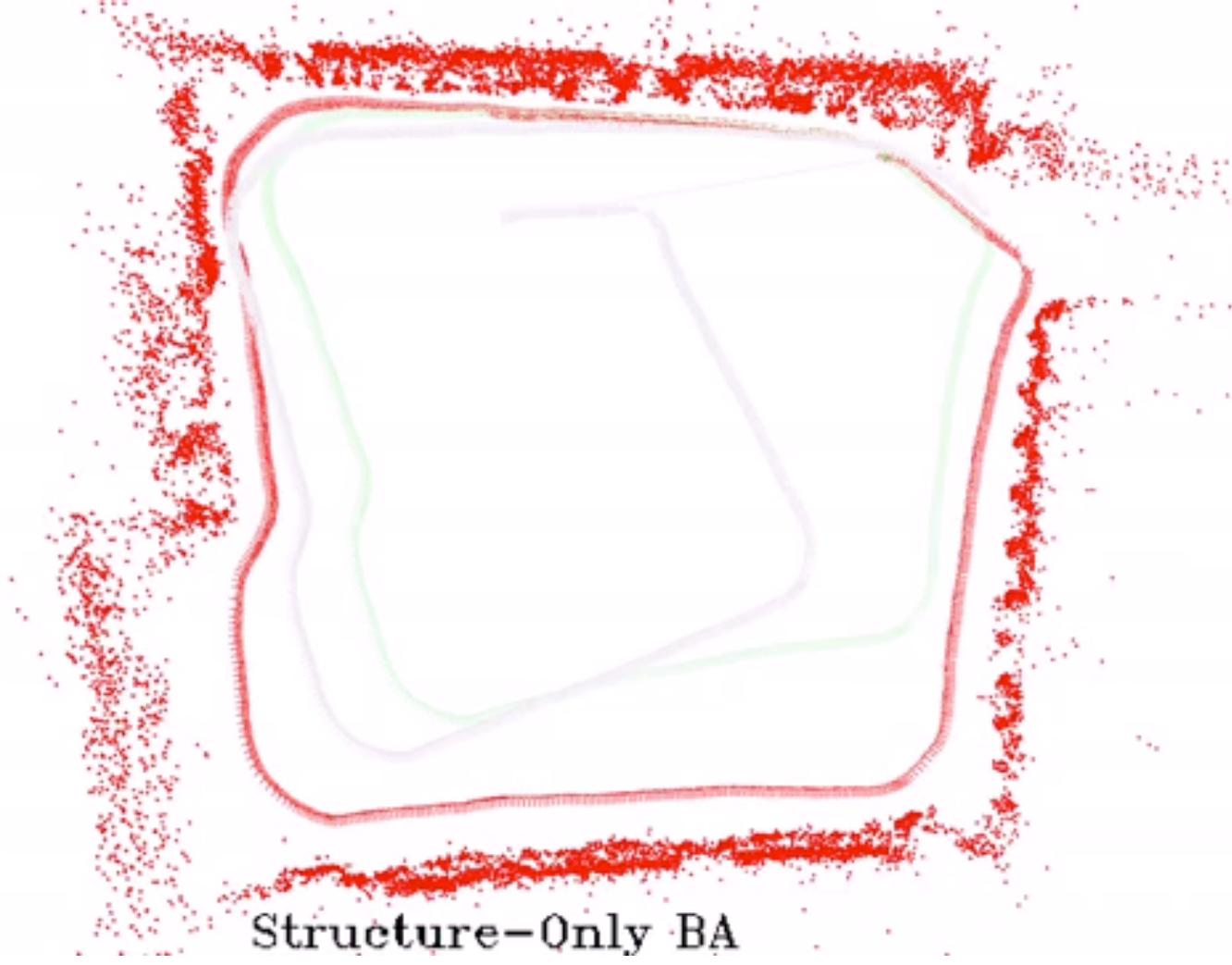
Map before optimisation

6-DOF optimization



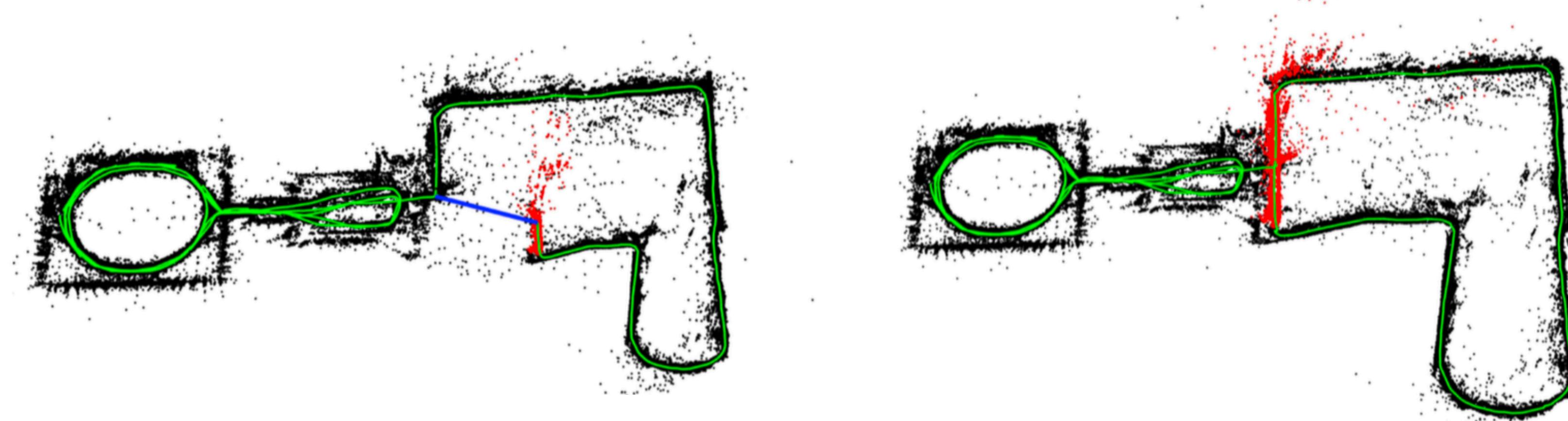
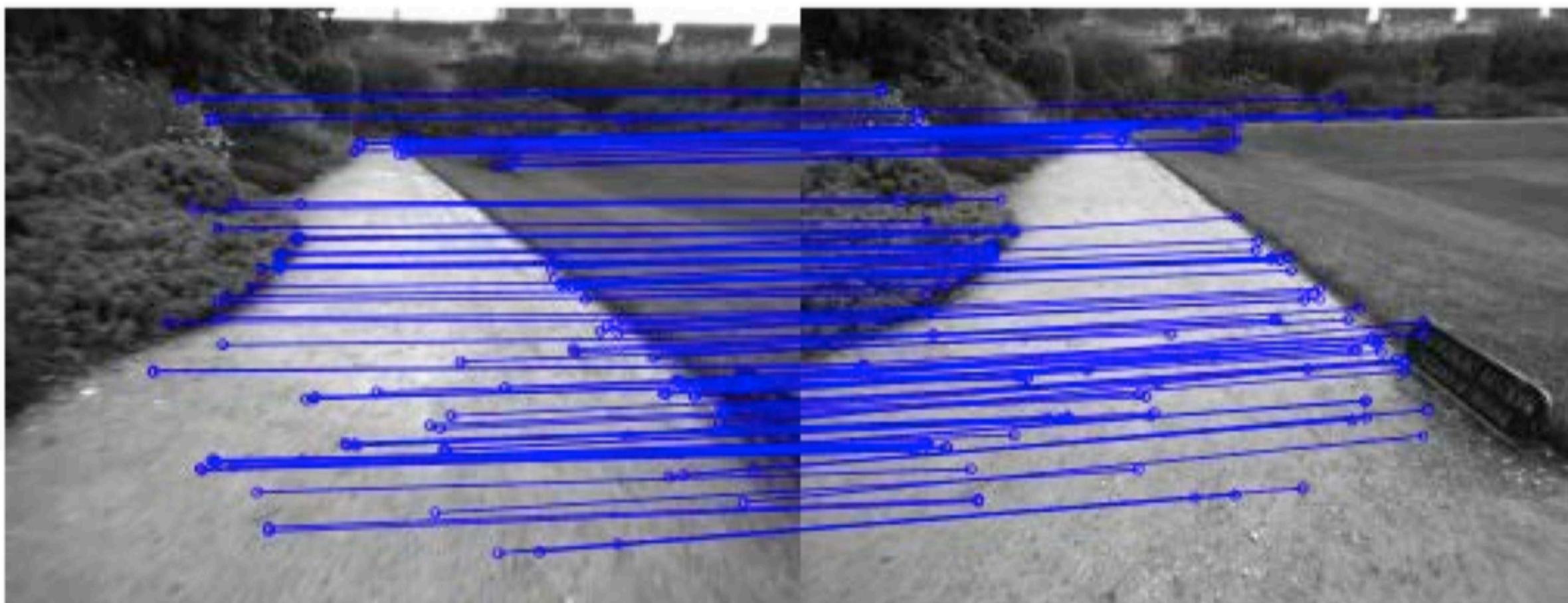
Projected points in

7-DOF optimization



Structure-Only BA

# Loop Correction



# Experiments

# On Outdoor Sequence (KITTI)

## ORB-SLAM

Raúl Mur-Artal, J. M. M. Montiel and Juan D. Tardós

{raulmur, josemari, tardos} @unizar.es



Instituto Universitario de Investigación  
en Ingeniería de Aragón  
**Universidad** Zaragoza



**Universidad**  
Zaragoza

# On Indoor Sequence (TUM RGB-D Benchmark)

## ORB-SLAM

Raúl Mur-Artal, J. M. M. Montiel and Juan D. Tardós

{raulmur, josemari, tardos} @unizar.es



Instituto Universitario de Investigación  
en Ingeniería de Aragón  
**Universidad** Zaragoza



**Universidad**  
Zaragoza

# Conclusions

A robust monocular SLAM system capable of operating for long durations in large-scale environments

Leverages parallel threads for tracking, local mapping and loop closing

Uses same ORB features for tracking, mapping, loop detection, relocalization

Introduces new key ideas in a monocular SLAM system,

- Covisibility and essential graphs
- DBoW2 based loop detection and relocalization
- Scale drift-aware loop correction

# Limitations and Extensions

## Limitations

- Monocular SLAM system, absolute scale unknown
- Needs texture, will fail with large plain walls
- Feature based maps too sparse a representation of environment

## Extensions

- Use stereo cameras: real scale and robustness to quick motions [1]
- Use RGB-D cameras: track with features + dense map [1]
- Add Semi-dense or dense mapping [2]
- Improve agility using IMU [3]
- Incorporating semantic information

[1] Mur-Artal, Raul, and Juan D. Tardós. "ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras." IEEE Transactions on Robotics 33.5, 2017: 1255-1262.

[2] Mur-Artal, Raúl, and Juan D. Tardós. "Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM." Robotics: Science and Systems. 2015.

[3] Mur-Artal, Raúl, and Juan D. Tardós. "Visual-inertial monocular SLAM with map reuse." IEEE Robotics and Automation Letters 2.2, 2017: 796-803.

# Feature-based vs Direct SLAM

	<b>Feature-Based SLAM (ORB-SLAM)</b>	<b>Direct SLAM (LSD-SLAM)</b>
<b>Matching</b>	Illumination/viewpoint invariance <i>Wide baseline</i>	Photometric consistency <i>Narrow baseline</i>
<b>Map Optimization</b>	Local BA + Global Pose Graph	Global Pose Graph
<b>Loop Detection</b>	Integrated Place Recognition	Need Features (FabMap)
<b># Points</b>	~300	~100K
<b>Strengths</b>	Excellent accuracy Robust in dynamic scenes Robust initialization	Robust in low texture areas Robust under defocus/motion blur
<b>Weakness</b>	Low texture areas Motion blur	Dynamic Objects Strong viewpoint/illumination changes Rolling-shutter cameras

Questions?