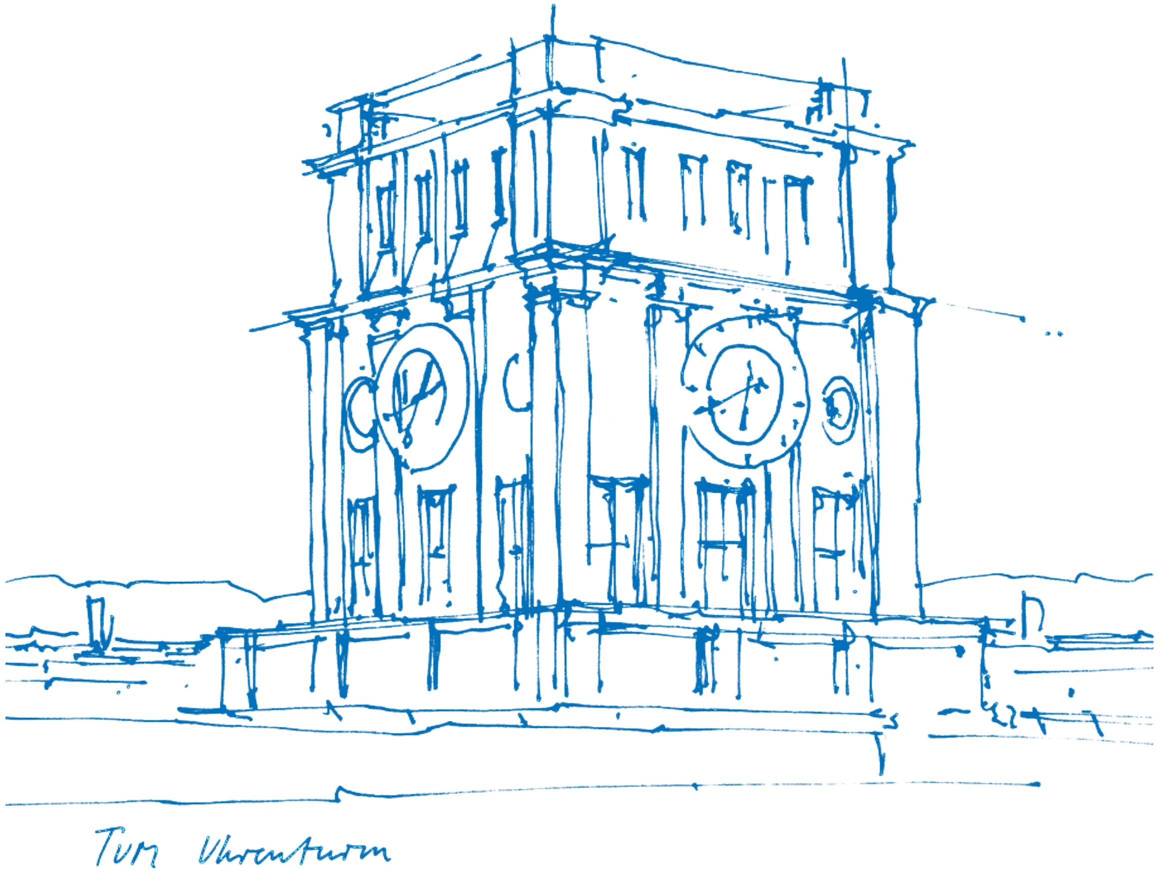
ORB-SLAM: A Versatile and Accurate Monocular SLAM System (2015)

Authors: Raul Mur-Artal, J.M.M. Montiel, Juan D. Tardos
Technische Universität München
Seminar: The Evolution of Motion Estimation and Real-time 3D Reconstruction
Supervisor: Christiane Sommer
Ezgi Çakır
15.12.2020



Outline

1. Introduction
2. Overview
3. Method Description
4. Experiments and Results
5. Summary
6. Comments

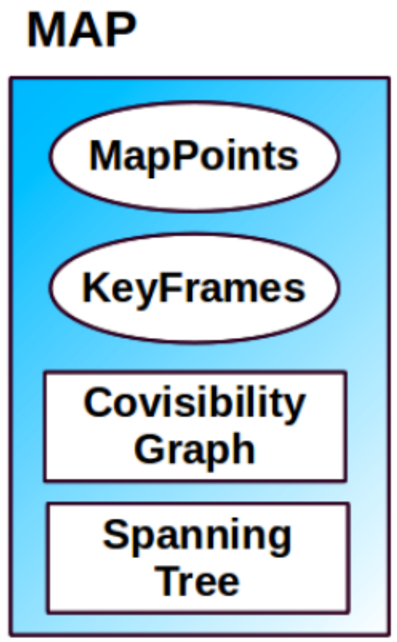
1. Introduction

- Feature-based monocular SLAM
- Real-time operation
- Unprecedented performance with respect to other state-of-the-art SLAM approaches

2. Overview

- **Contributions**
  - Use same ORB features for tracking, mapping, relocalization, and loop closing
    - real-time without GPU, invariance to changes
  - Tracking and mapping on local covisible area (Covisibility Graph)
  - Capability of relocalization, recovery from failure, enhanced map reuse
  - Loop closing based on optimization of a pose graph (Essential Graph)
  - An automatic and robust initialization procedure based on model selection among planar and non-planar scenes
  - A *survival of the fittest* approach to map point and keyframe selection, discard redundant keyframes —> improved tracking robustness

2. Overview

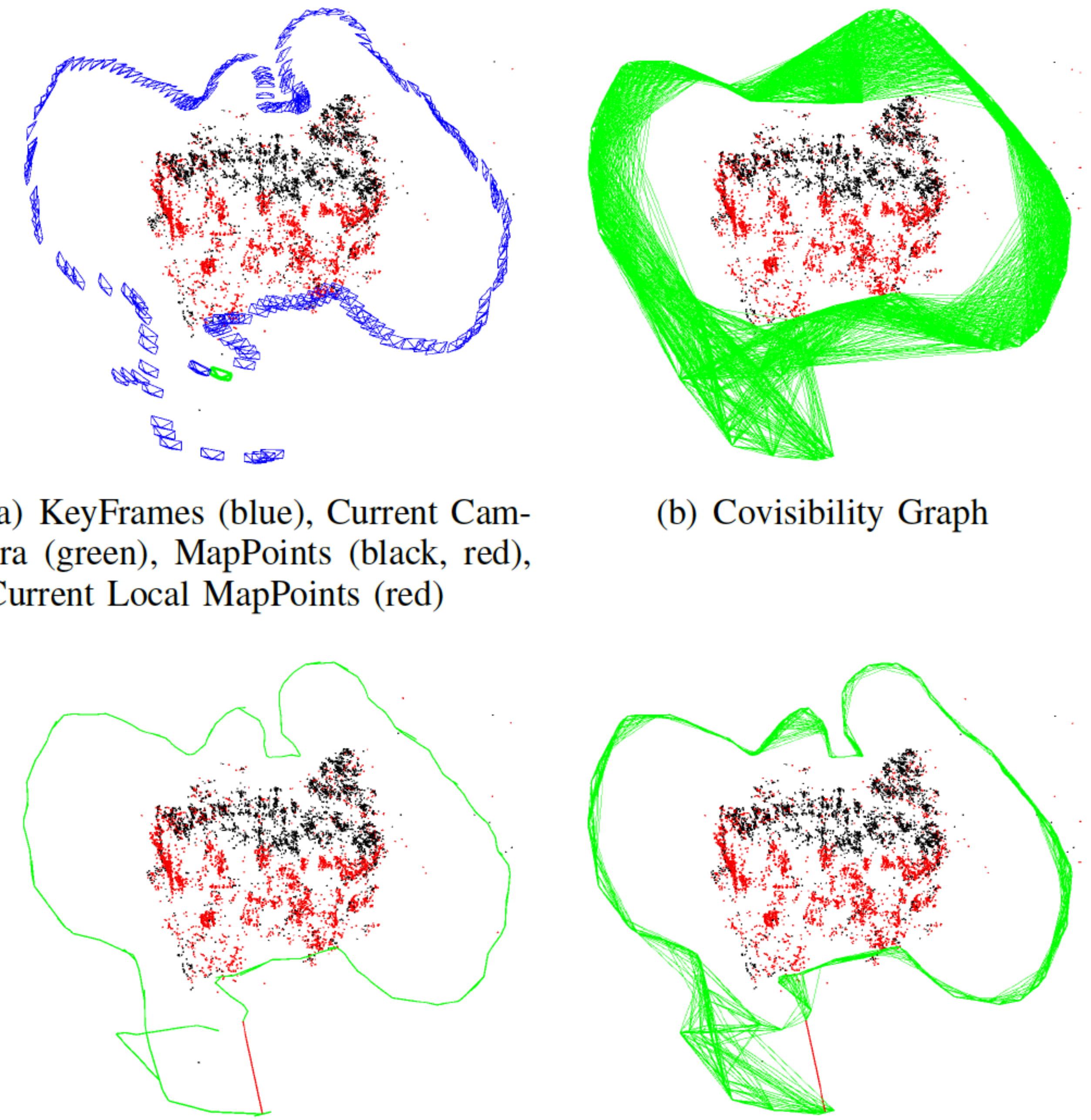


Covisibility Graph

- Undirected weighted graph
- Very dense
- θ >= 15

Essential Graph

- Spanning tree
- All nodes, subset of edges and loop closure edges
- θ >= 100
- Preserves strong network



(a) KeyFrames (blue), Current Camera (green), MapPoints (black, red), Current Local MapPoints (red)

(b) Covisibility Graph

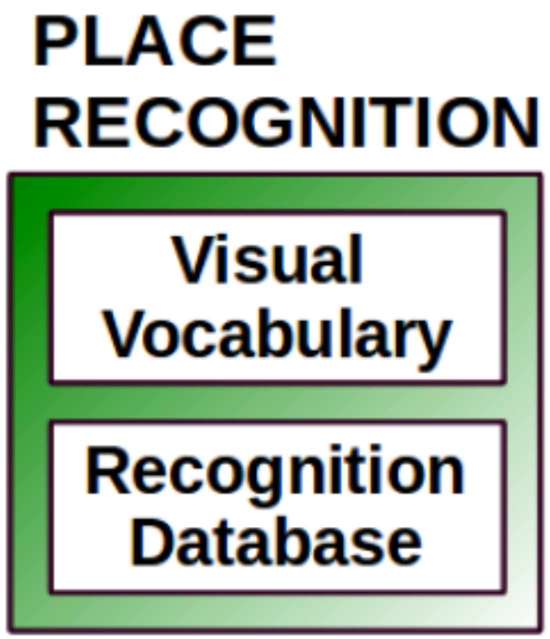
(c) Spanning Tree (green) and Loop Closure (red)

(d) Essential Graph

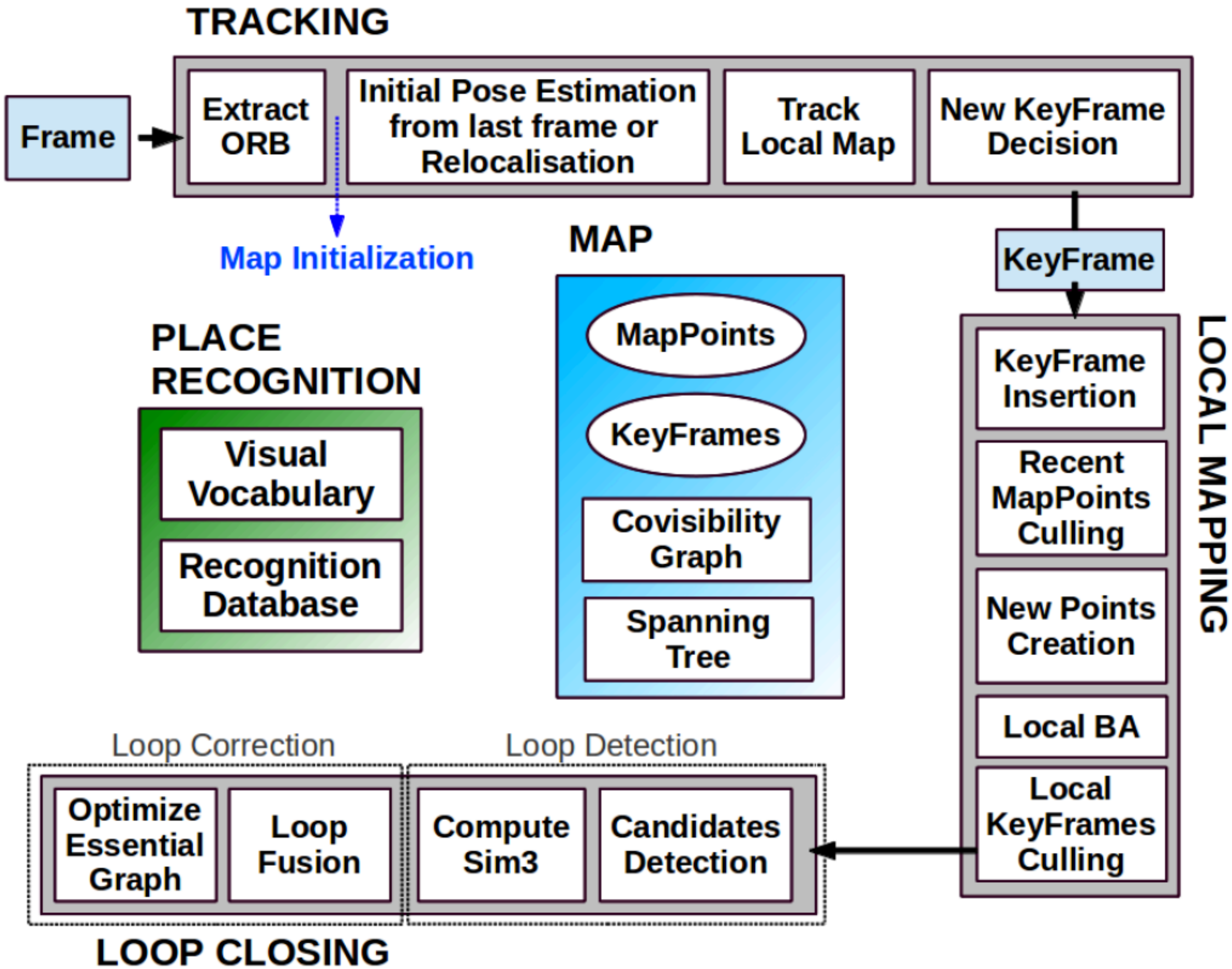
Overview

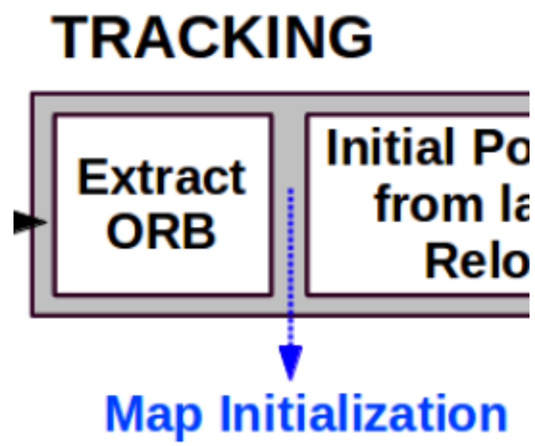
- Bag of Words Place Recognition
  - based on DBoW2 to perform loop detection and relocalization

- Vocabulary created offline with ORB descriptors



2. Method description





Map Initialization

Goal: Compute relative pose and triangulate an initial set of map points

1. Extract ORB features and match

2. Compute 2 geometrical models in parallel
   - Homography matrix & Fundamental matrix
   - Compute model score

$$\mathbf{x}_c = \mathbf{H}_{cr}\,\mathbf{x}_r$$

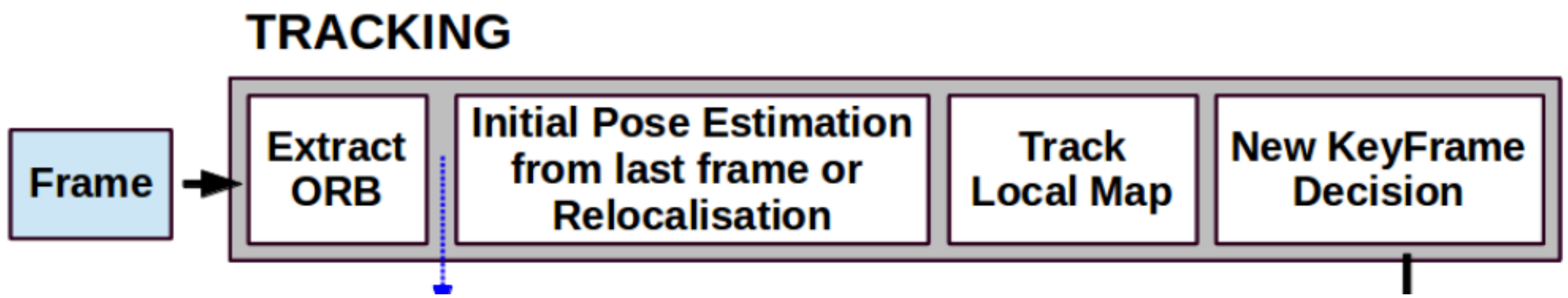
$$\mathbf{x}_c^T\,\mathbf{F}_{cr}\,\mathbf{x}_r = 0$$

3. Use heuristic, select model

4. Try to recover the relative pose
   - Not enough inliers, then back to 1
     - ✓ Robust under low-parallax and the twofold ambiguity configuration

5. Full bundle adjustment

Tracking



1. Extract ORB features

2. Initial pose estimation if previously successful tracking
    - Matches features & optimize poses

3. Initial pose estimation if tracking is lost
    - Global relocalization with bag of words
    - Find pose & optimize

4. Track the local map
    - Local map: set of keyframes, their neighbors and seen map points
    - Project the map points and match & discard some map points

5. Decide whether the current frame stays as a keyframe or not
    - Insert as fast as possible
        - ✓ Robust to camera movements

Local Mapping

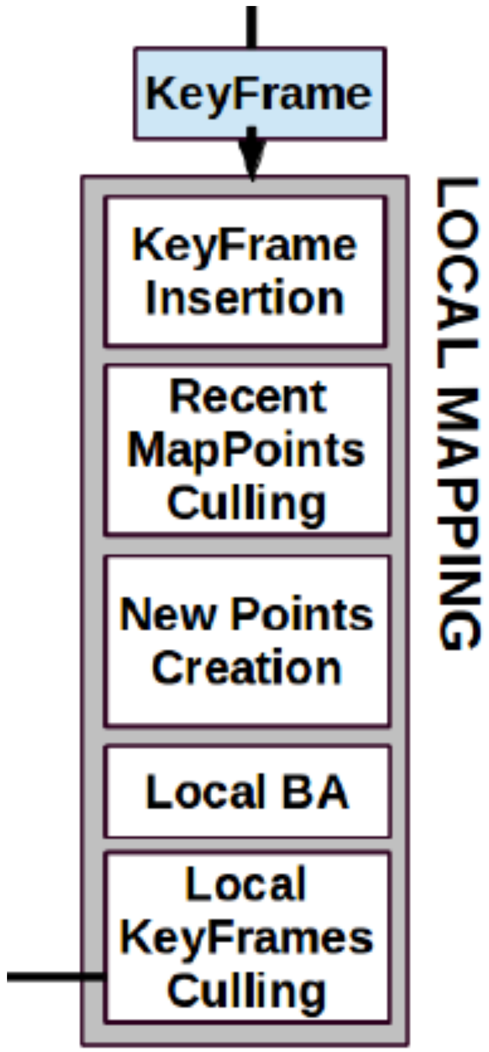
1. Insert the keyframe
   - Update covisibility graph & BoW representation

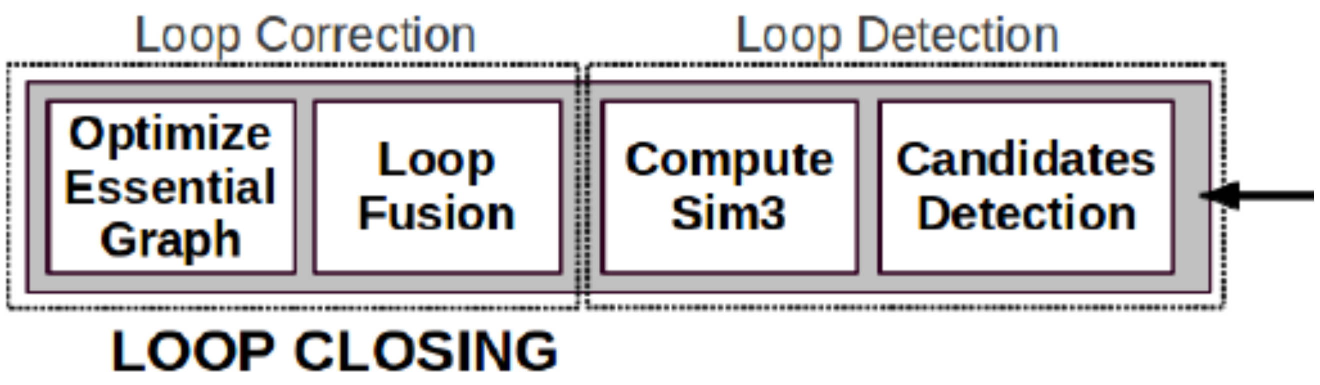
2. Cull recent map points

3. New point creation
   - Triangulate ORB feature & match
   - Discard if not fulfill the epipolar constraint

4. Local bundle adjustment

5. Cull local keyframes





Loop Closing

1. Detect loop candidates
   - Candidates from the non-neighbors and with score more than the threshold

2. Compute the similarity transformation (7 DoF)
   - Information about the error accumulated and serve as geometrical validation

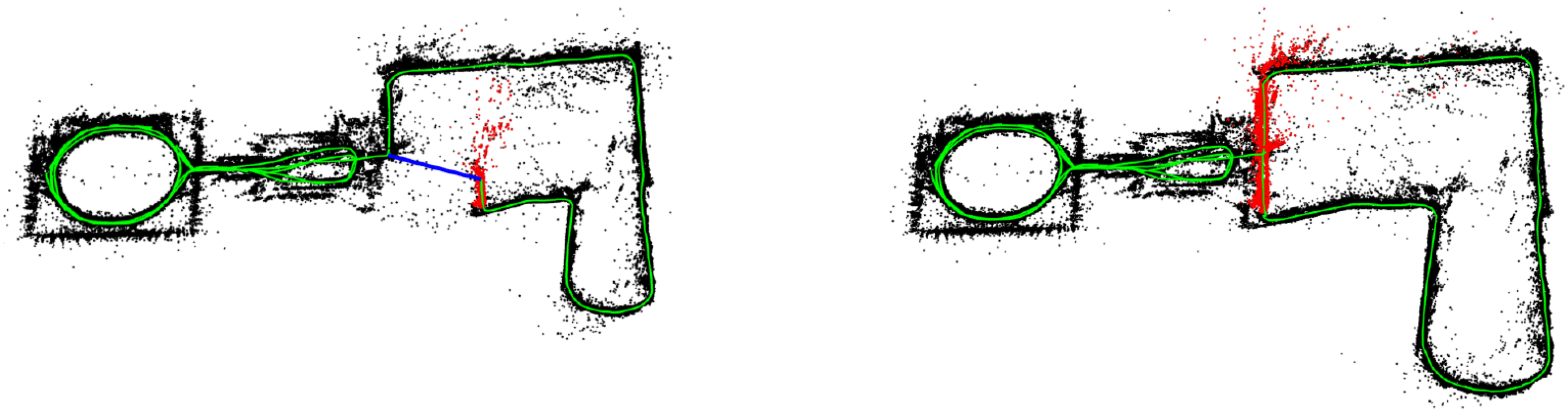
3. Loop fusion
   - Correct the keyframe pose & propagate
   - Update the covisibility graph

4. Optimize the essential graph
   - Pose graph optimization
   - Distribute the loop closing error along the graph

4. Experiments and Results

1. NewCollege: large robot sequences
2. TUM RGB-D benchmark: 16 indoor scenes
3. KITTI: 10 car outdoor sequences

4. Experiments and Results



Map before and after a loop closure in the NewCollege sequence

4. Experiments and Results

TABLE II
LOOP CLOSING TIMES IN NEWCOLLEGE

Loop	KeyFrames	Essential Graph Edges	Loop Detection (ms)		Loop Correction (s)		Total (s)
			Candidates Detection	Similarity Transformation	Fusion	Essential Graph Optimization	
1	287	1347	4.71	20.77	0.20	0.26	0.51
2	1082	5950	4.14	17.98	0.39	1.06	1.52
3	1279	7128	9.82	31.29	0.95	1.26	2.27
4	2648	12547	12.37	30.36	0.97	2.30	3.33
5	3150	16033	14.71	41.28	1.73	2.80	4.60
6	4496	21797	13.52	48.68	0.97	3.62	4.69

- Time needed for loop closing increases sublinearly with the number of keyframes.
- Reason: BoW & sparse essential graph

4. Experiments and Results

Localization accuracy in terms of Absolute Trajectory Error

- ORB-SLAM is able to process most of the sequences
- Higher accuracy when detecting large loops.

TABLE III
KEYFRAME LOCALIZATION ERROR COMPARISON IN THE TUM RGB-D BENCHMARK [38]

	Absolute KeyFrame Trajectory RMSE (cm)			
	ORB-SLAM	PTAM	LSD-SLAM	RGBD-SLAM
fr1_xyz	**0.90**	1.15	9.00	1.34 (1.34)
fr2_xyz	0.30	**0.20**	2.15	2.61 (1.42)
fr1_floor	**2.99**	X	38.07	3.51 (3.51)
fr1_desk	**1.69**	X	10.65	2.58 (2.52)
fr2_360 _kidnap	3.81	**2.63**	X	393.3 (100.5)
fr2_desk	**0.88**	X	4.57	9.50 (3.94)
fr3_long _office	**3.45**	X	38.53	-
fr3_nstr_ tex_far	ambiguity detected	4.92 / 34.74	18.31	-
fr3_nstr_ tex_near	**1.39**	2.74	7.54	-
fr3_str_ tex_far	**0.77**	0.93	7.95	-
fr3_str_ tex_near	1.58	**1.04**	X	-
fr2_desk _person	**0.63**	X	31.73	6.97 (2.00)
fr3_sit_ xyz	**0.79**	0.83	7.73	-
fr3_sit_ _halfsph	**1.34**	X	5.87	-
fr3_walk _xyz	**1.24**	X	12.44	-
fr3_walk _halfsph	**1.74**	X	X	-

4. Experiments and Results

Relocalization comparing with PTAM using TUM RGB-D benchmark

- ORB-SLAM accurately relocalizes more than the double of frames than PTAM.

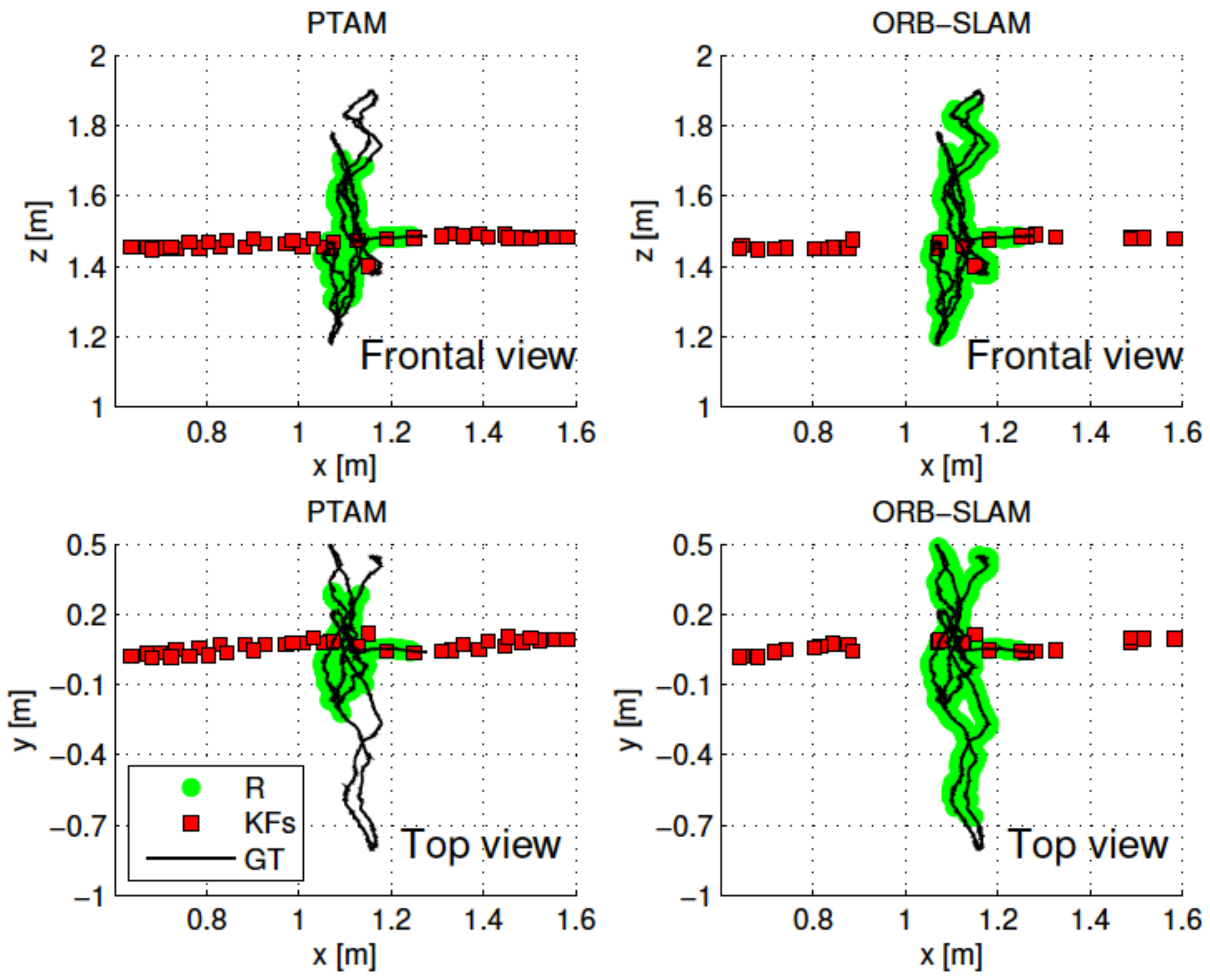
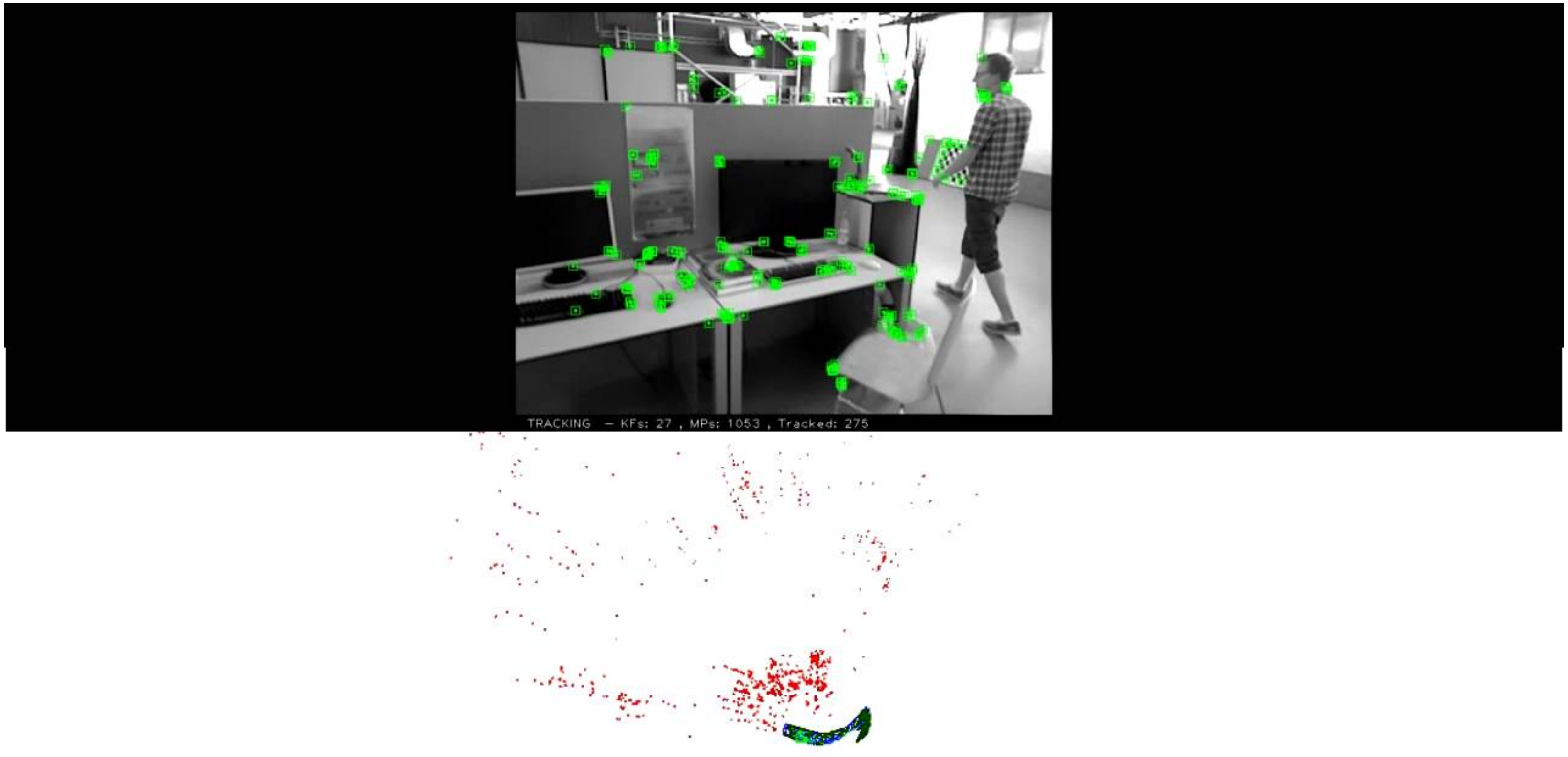


Fig. 7. Relocalization experiment in *fr2_xyz*. Map is initially created during the first 30 seconds of the sequence (KFs). The goal is to relocalize subsequent frames. Successful relocalizations (R) of our system and PTAM are shown. The ground truth (GT) is only shown for the frames to relocalize.

4. Experiments and Results

- Relocalization is robust under dynamic cases.
- Even in dynamic sequences, ORB-SLAM relocalizes 78% of the frames.

Fig. 8. Example of challenging relocalizations (severe scale change, dynamic objects) that our system successfully found in the relocalization experiments.



Example of a challenging dynamic scene

4. Experiments and Results

- The number of keyframes saturates.
- Culling procedure of keyframes helps.

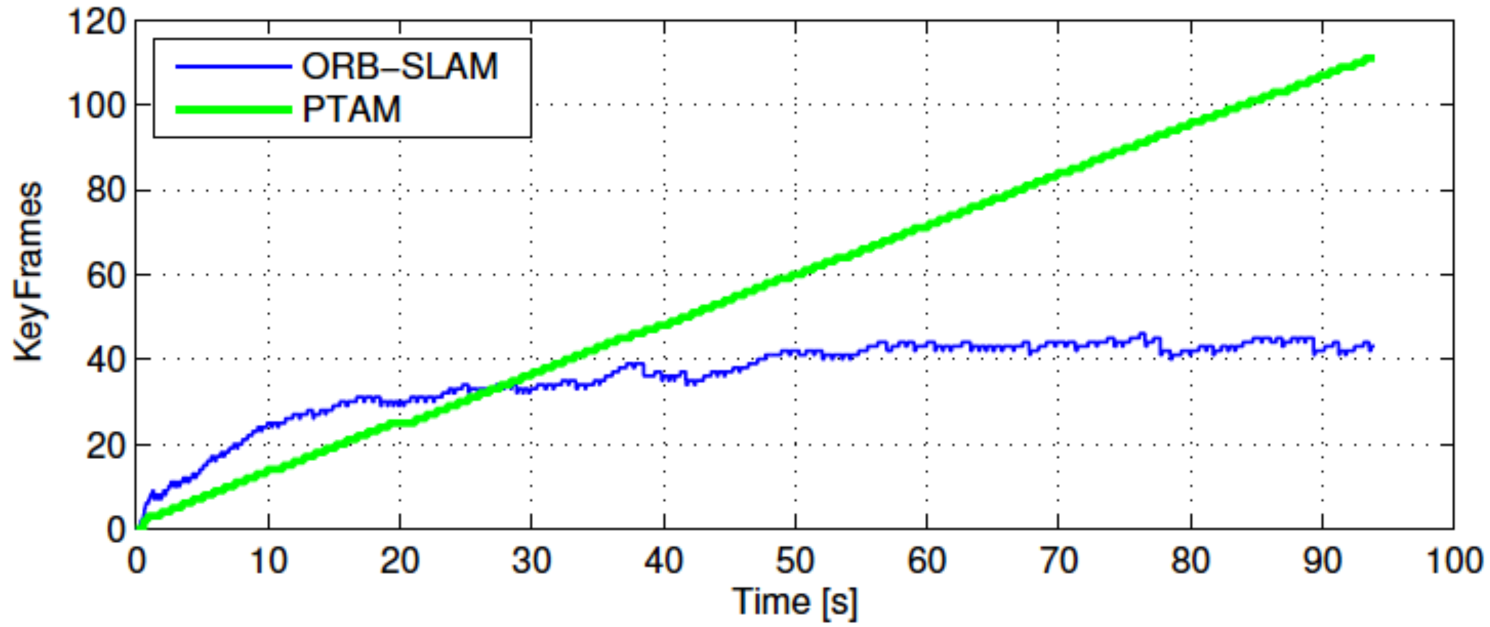


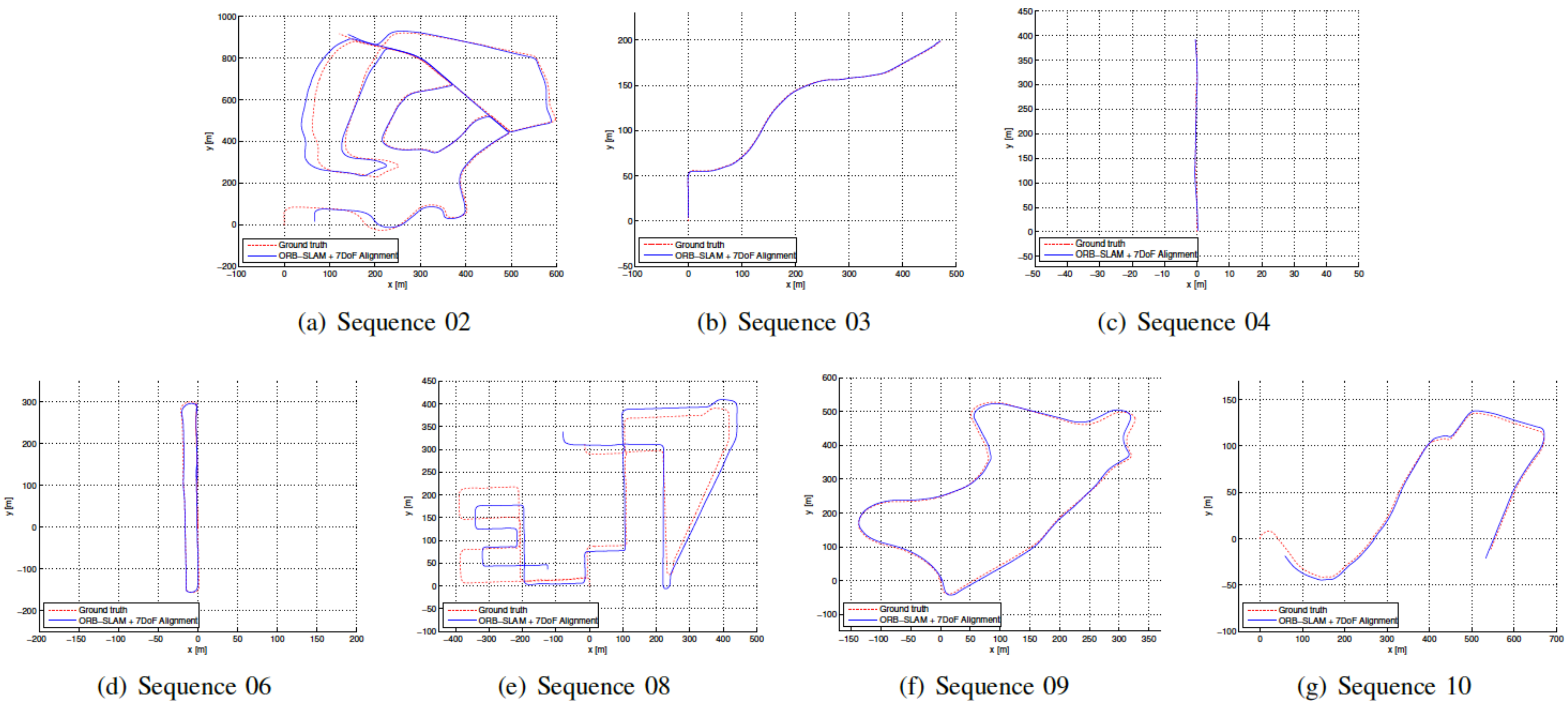
Fig. 9. Lifelong experiment in a static environment where the camera is always looking at the same place from different viewpoints. PTAM is always inserting keyframes, while ORB-SLAM is able to prune redundant keyframes and maintains a bounded-size map.

4. Experiments and Results

Loop Closing in the KITTI Dataset

- Accurate trajectories, exception of sequence 08

- ORB-SLAM needs loop closure for accurate reconstruction.



(a) Sequence 02 (b) Sequence 03 (c) Sequence 04

(d) Sequence 06 (e) Sequence 08 (f) Sequence 09 (g) Sequence 10

Fig. 12. ORB-SLAM keyframe trajectories in sequences 02, 03, 04 ,06, 08, 09 and 10 from the odometry benchmark of the KITTI dataset. Sequence 08 does not contains loops and drift (especially scale) is not corrected.

6. Summary

- Indoor and outdoor scenes

- The accuracy is impressive.

- Combined new and old ideas,
  such as the loop detection, the loop closing procedure and covisibility graph, and ORB features.

- Adding new frames soon, cull when redundant
  —> capture fast movement while maintaining a compact but representative graph

5. Comments

- 👍🏻 Essential Graph, Covisibility Graph, BoW
- 👍🏻 Not overgrowing of map and number of frames in graph
- 👍🏻 Invariance and robust to different viewpoints
- 👍🏻 Applicable both indoor and outdoor scenes
- 👍🏻 Parallel computing of building blocks
- 👎🏻 Need of loop closure

Thank you for your attention.

Questions?