SLAM

3D Vision

University of Illinois

Derek Hoiem

SLAM example

https://www.youtube.com/watch?v=2dLWqv37sEE

Key concepts/steps:

1. Frame-to-frame tracking (visual odometry)
2. Frames, KeyFrames, Point Maps
3. Local BA, Full BA
4. Relocalization and Loop Closure

SfM and SLAM: similarities

- Solve for camera poses and 3D scene points given images
- Correspondence, registration, outlier rejection, and bundle adjustment are core problems

SfM vs. SLAM: differences

SfM	SLAM
• Input is unordered set of images	• Input is stream of images, stereo, or depth and sometimes IMU
• Focus is on precision, with aim to produce a good 3D model	• Focus is on speed and robustness, with aim to localize camera or robot
• Offline, one-time process	• Online process, possibly with relocalization
• Published mainly in vision conferences	• Published mainly in robotics conferences
• 3 papers with more than 1000 citations	• 8 papers with more than 1000 citations
• Complicated	• Very complicated

This class: SLAM

- ORB-SLAM and extensions
  - ORB-SLAM 1: technical walkthrough
  - Break
  - Results of ORB-SLAM 1
  - Summary of improvements in ORB-SLAM 2 and ORB-SLAM 3

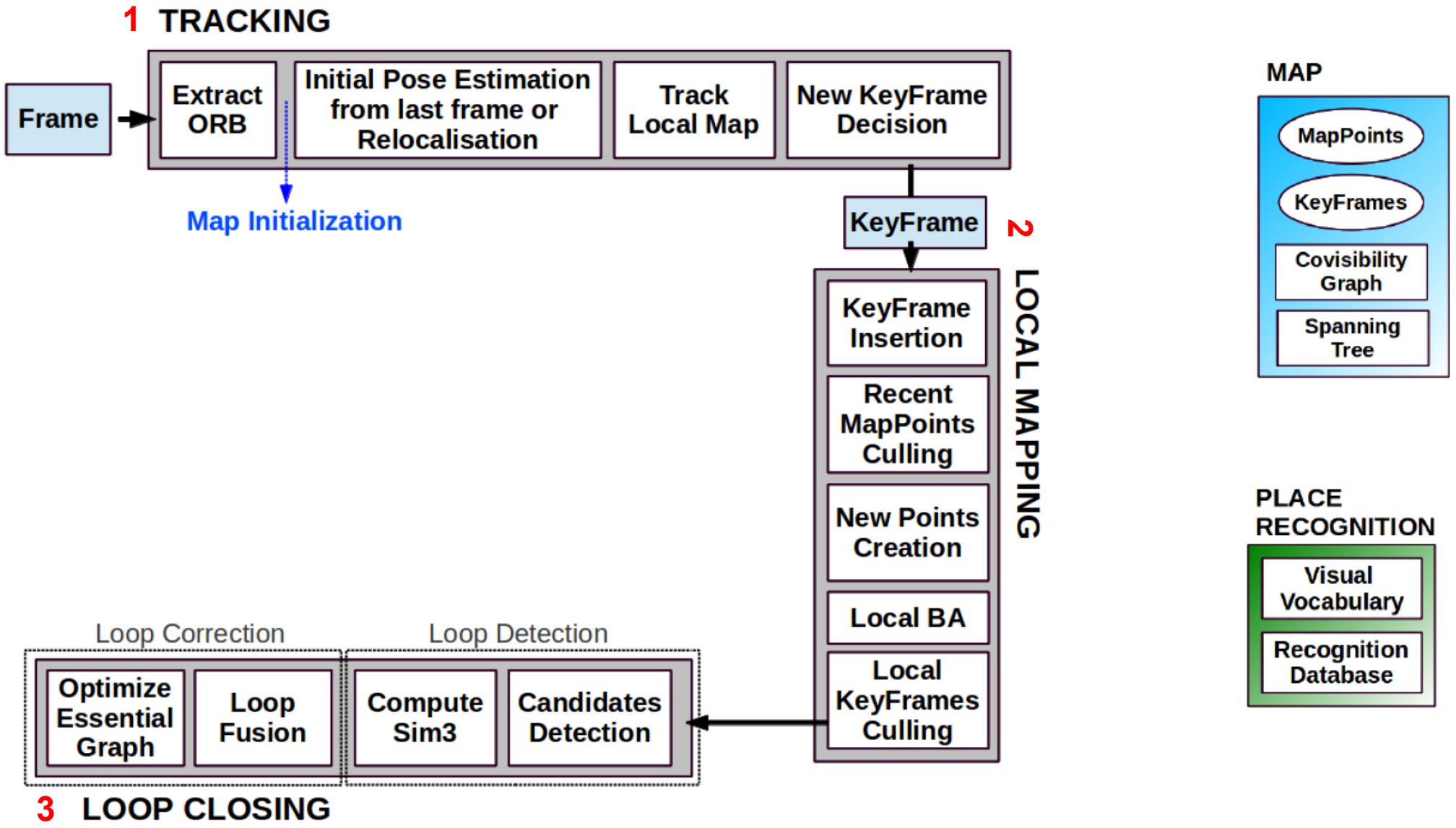
- LSD-SLAM

IEEE TRANSACTIONS ON ROBOTICS 2015

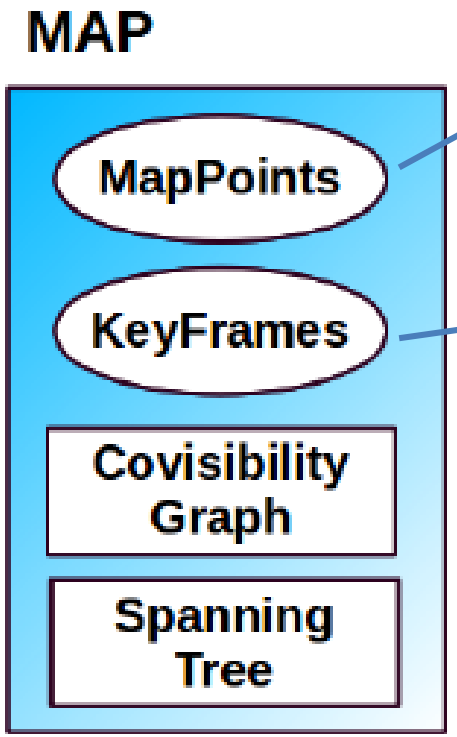
ORB-SLAM: a Versatile and Accurate Monocular SLAM System

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

ORB-SLAM: three parallel threads

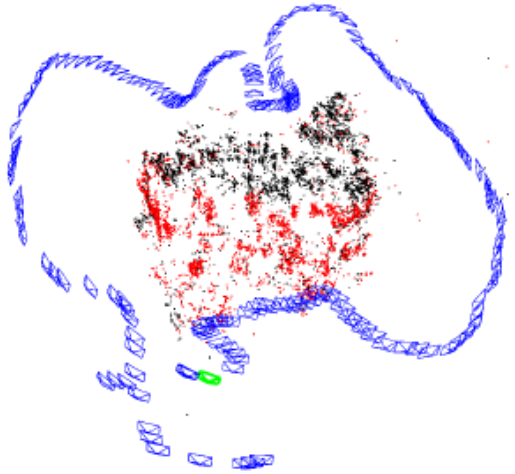


ORB-SLAM data

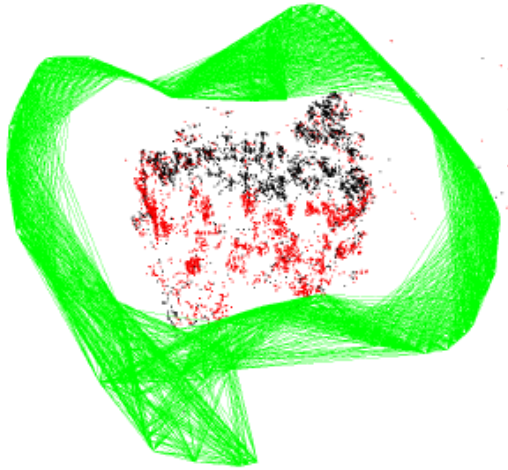


- 3D position ($\mathbf{X}_{w,i}$)
- average viewing direction $\mathbf{n}_i$
- centroid ORB descriptor $\mathbf{D}_i$
- Observable distance range

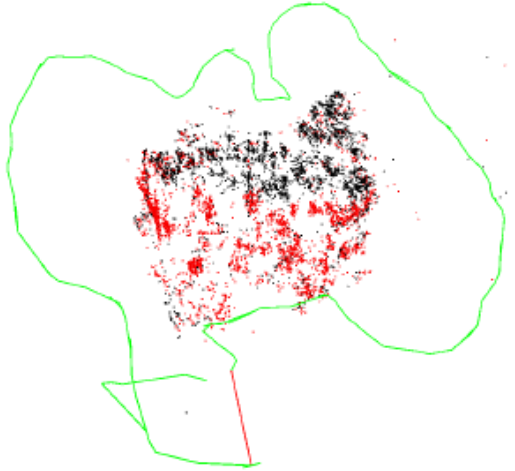
- Camera pose $\mathbf{T}_{iw}$
- All ORB features



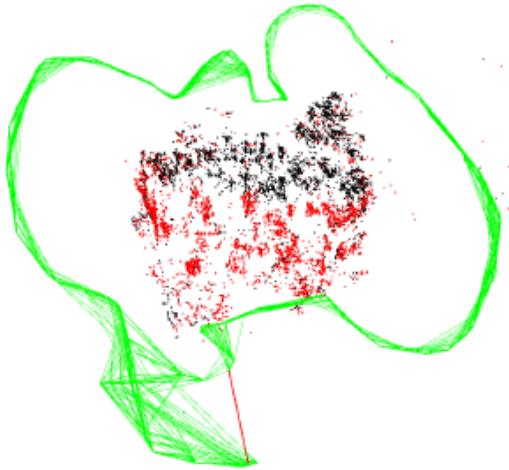
(a) KeyFrames (blue), Current Camera (green), MapPoints (black, red), Current Local MapPoints (red)



(b) Covisibility Graph

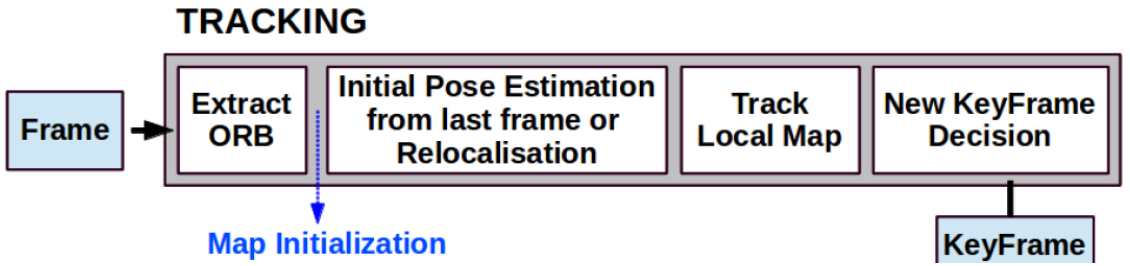


(c) Spanning Tree (green) and Loop Closure (red)



(d) Essential Graph

Tracking Overview



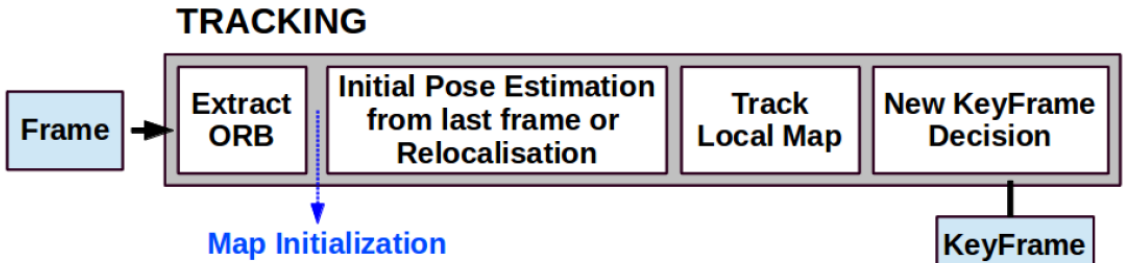
Goal: Achieve fast and robust matching of each frame based on points observed in last frame

For each new frame

1. Extract features
2. Localize to previous frame (if possible) or keyframes
3. Find more matching points
4. Potentially add this frame to set of keyframes

This is “visual odometry” and could also be augmented with IMU (intertial sensor)

Tracking: Map Initialization

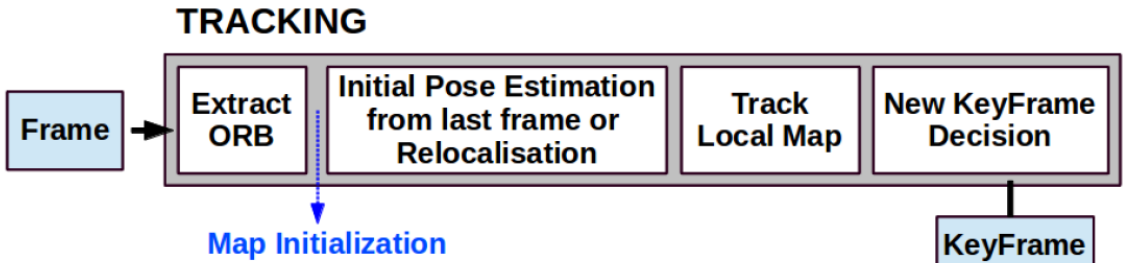


- Extract ORB features at finest scale
- Find correspondences in two frames (current and reference)
- Check whether points are mostly explained by homography H or fundamental matrix F
  - If F: solve for F and compute E using intrinsic matrix K

$$\mathbf{E}_{rc} = \mathbf{K}^T \mathbf{F}_{rc} \mathbf{K}$$

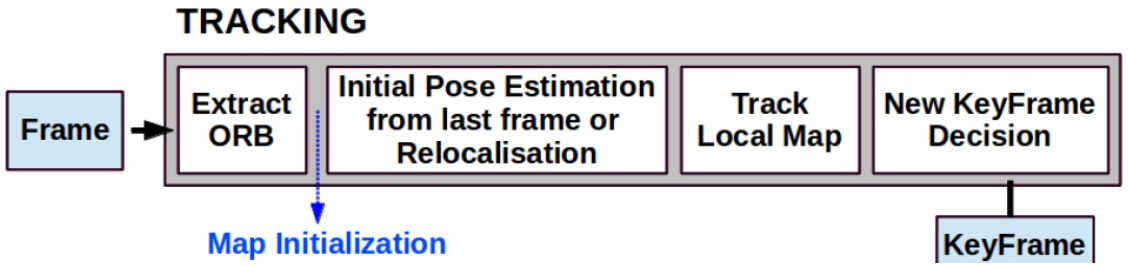
  - If H: check for valid planar solution
  - Get a new reference frame if well-conditioned F or H cannot be found
- Perform bundle adjustment (BA) on two frames and mapped points

Tracking: ORB + Initial Pose



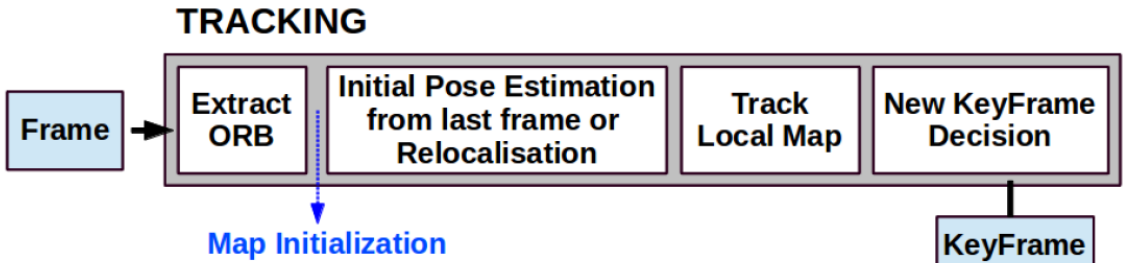
- FAST corners + ORB features
  - 1000-2000 corners
  - Distributed across 8 scales and a grid of cells
- Try to initialize pose using previous frame
  - Predict positions of previously observed map points into current frame based on constant velocity motion estimate
  - Perform wider search if not enough points found
- Else, perform relocalization
  - Find candidate matches among existing keyframes with bag of words search
  - Use RANSAC and PnP to optimize pose and then perform guided search of for more map points

Tracking: Track Local Map



- For keyframes that share map points and their neighbors, get all map points
- Find more map point correspondences
  1. Project each to current frame, check if in bounds
  2. Check that current view angle is within 60 deg of avg for point
  3. Check that current distance is within point range
  4. Find best matching ORB feature at similar position/scale, and associate
- Optimize camera pose wrt associated points

Tracking: Keyframe Decision



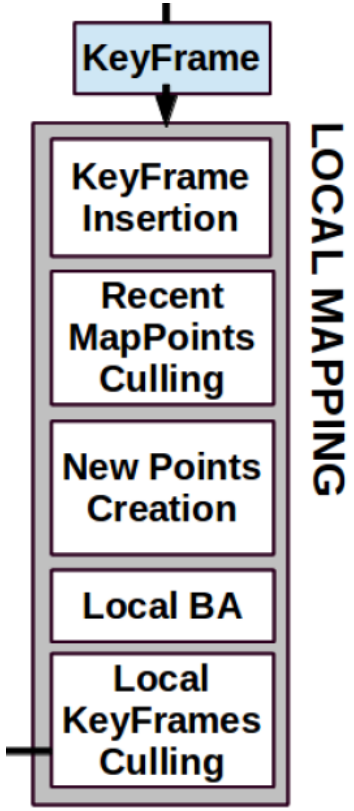
Add current frame as new keyframe K_i if all conditions are met:

1. More than 20 frames since last global relocalization
2. Ready for new: Local mapping idle, or more than 20 frames since last keyframe insertion
3. Good tracking: Current frame tracks at least 50 points
4. Not redundant: Current frame tracks less than 90% of points from most similar keyframe

Local Mapping Overview

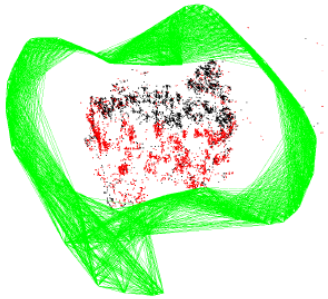
Goal: Jointly refine points and poses of recently viewed parts of the scene to reduce drift

1. Update graphs and maps with new keyframe K_i
2. Optimize nearby keyframes and points
3. Remove bad points and redundant keyframes

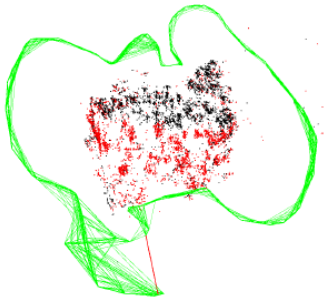


Local Mapping: Keyframe Insertion

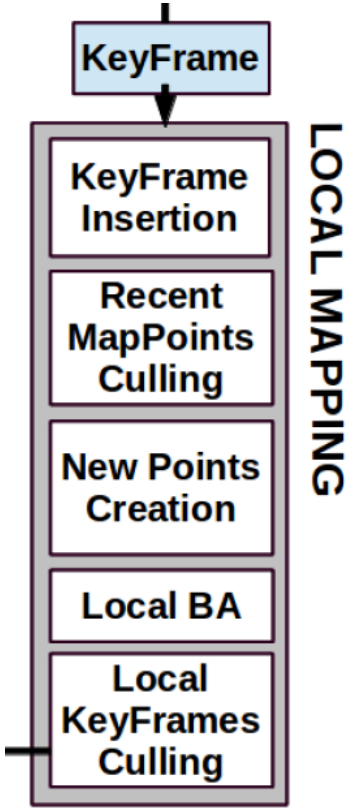
- Update covisibility graph (which other keyframes see the same points as K_i)
- Update spanning tree, linking with keyframe that has most points in common with K_i
- Compute bag of words for K_i



(b) Covisibility Graph



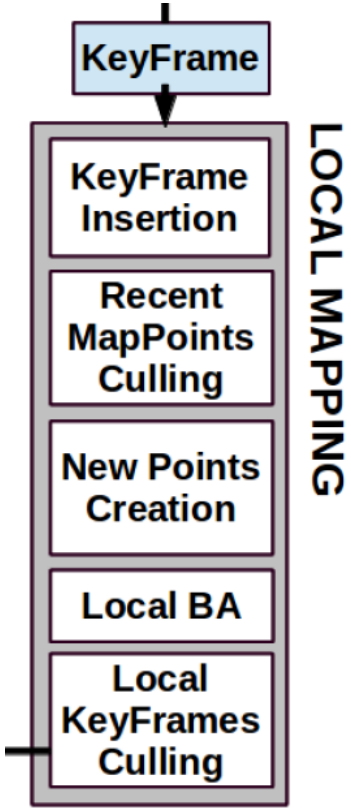
(d) Essential Graph



Local Mapping: Recent Point Culling

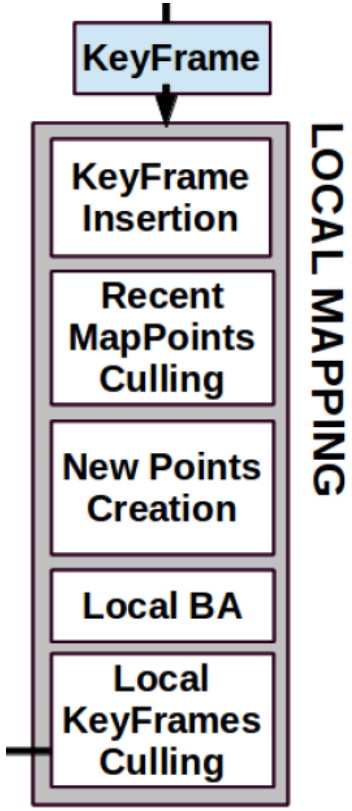
- Points are initially kept if
  - Point is tracked in at least 25% of expected frames
  - Observed in at least three keyframes (after initialization)

- Remove points if not enough keyframes (after keyframes removed), or high reprojection error after Local BA



Local Mapping: Point Creation

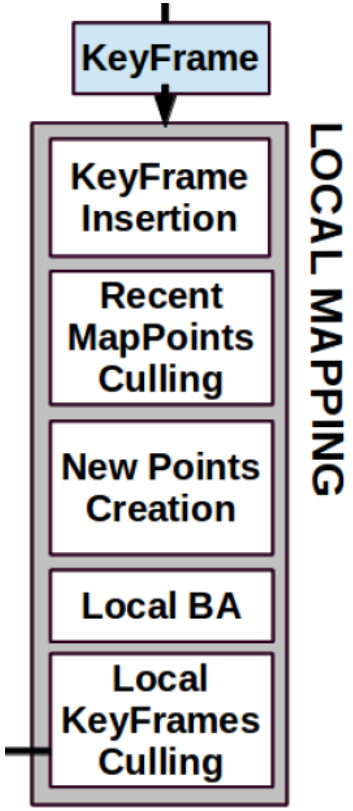
- Attempt to match any unmatched features in K_i to co-visible keyframes
  - Use vocab tree and check epipolar constraint
- Triangulate good matches, check reprojection error, etc.
- Find correspondences in additional connected keyframes, similar to “track local map”



Local Mapping: Local BA

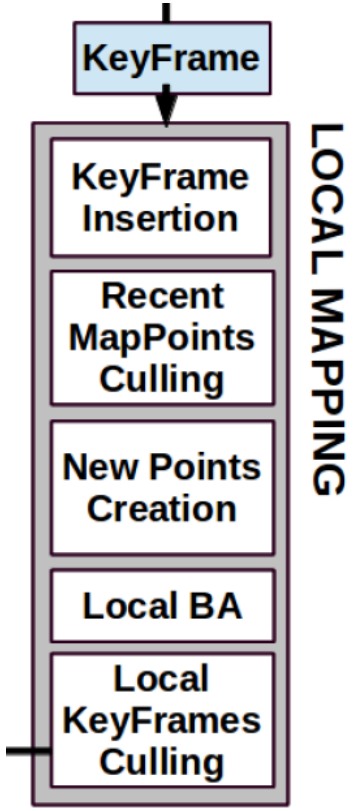
- Optimize new keyframe K_i, those connected in covisibility graph, and points seen by those keyframes

- Discard points that have high reprojection error at middle and end of process

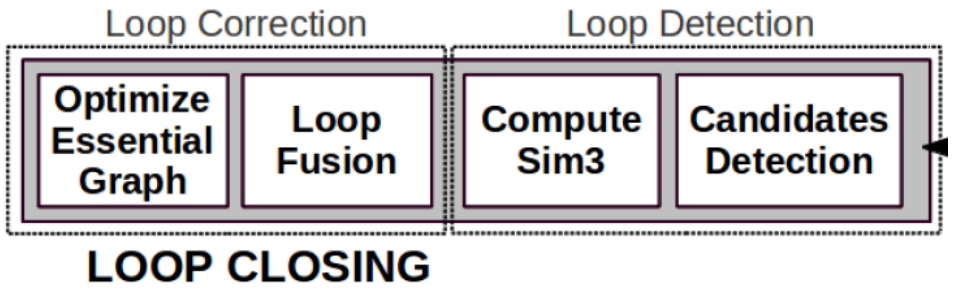


Local Mapping: KeyFrames culling

- Discard keyframes if at least 90% of its observed points are also observed by other keyframes at similar or closer distance



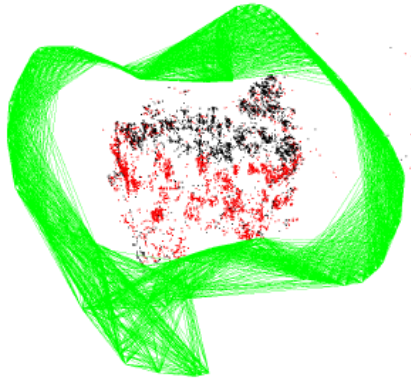
Loop Closing Overview



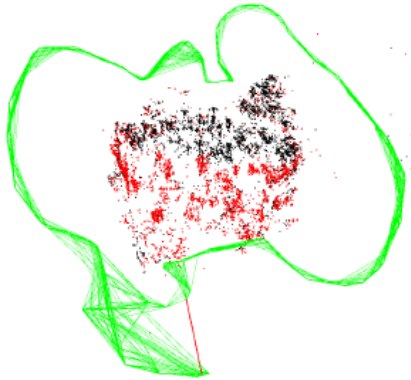
Goal: Find and optimize over long-range connections to eliminate drift

For new keyframe K_i:

1. Vocab tree matching to all non-connected keyframes and get candidates
2. Find matches with candidates and use RANSAC to solve similarity transform
3. Add edges to co-visibility and essential graph and perform graph optimization on essential graph

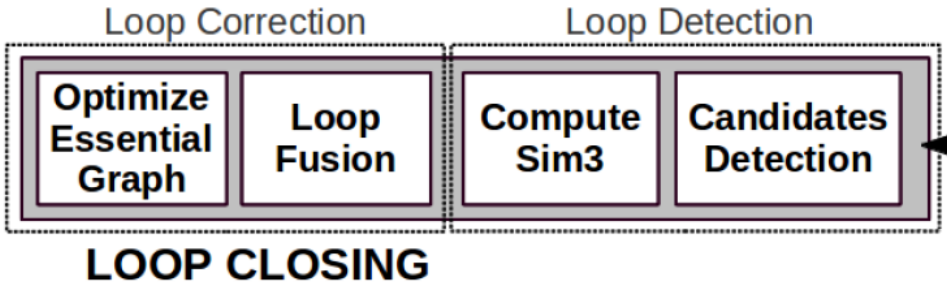


(b) Covisibility Graph

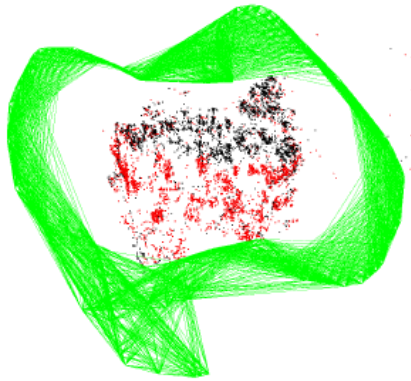


(d) Essential Graph

Loop: Candidate Detection

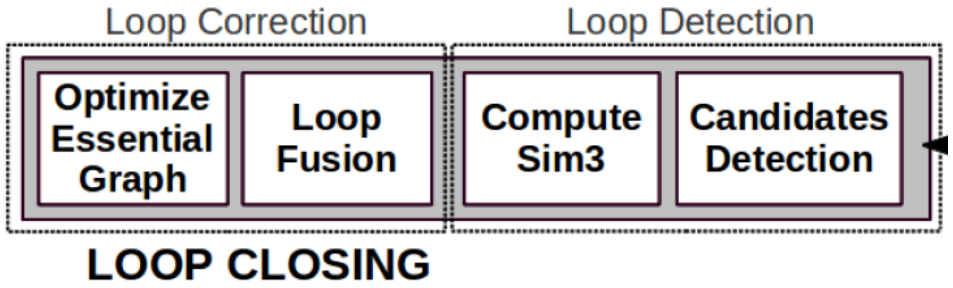


- Find BoW similarity of K_i to neighbors covisibility graph and set threshold S as minimum similarity
- Candidates are keyframes that
  - Are not connected to K_i
  - Have score greater than S
  - Two other connected keyframes in co-visibility graph also have score with K^i greater than S



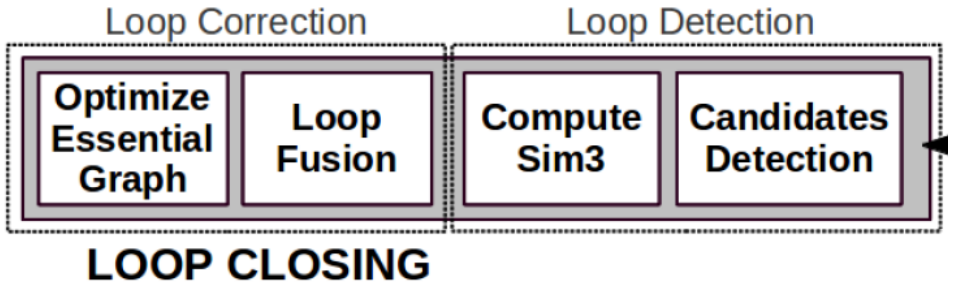
(b) Covisibility Graph

Loop: Similarity Transform

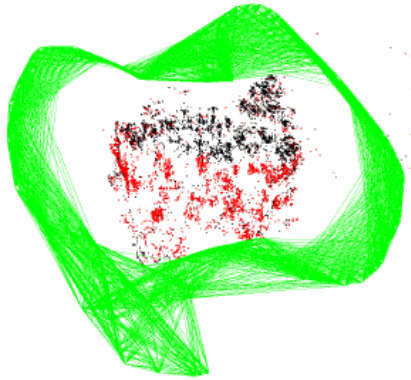


- Find feature matches between K_i and loop keyframe K_l
  - Provides 3D to 3D correspondences since features are linked to 3D points
- Solve for similarity transform with RANSAC
- Optimize camera pose with points fixed and perform guided search for more matches
- Accept loop closure with K_l if there are enough inliers

Loop Closing: Fusion

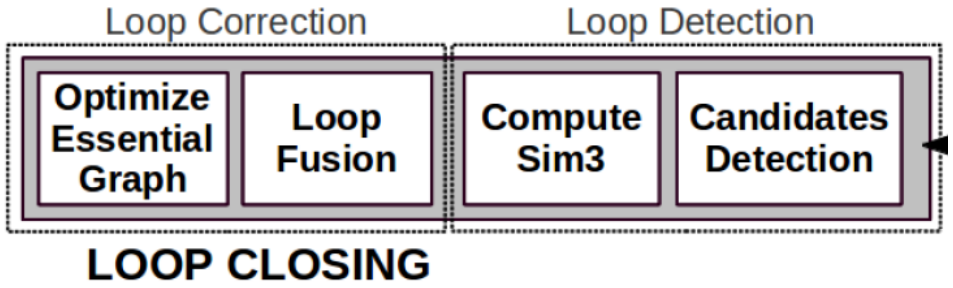


- Update pose of K_i and its neighbors with similarity transform
- Fuse points with K_i that are also seen by K_l and its neighbors
- Search for additional points to fuse by checking projections and feature similarities
- Update covisibility graph to reflect fused points



(b) Covisibility Graph

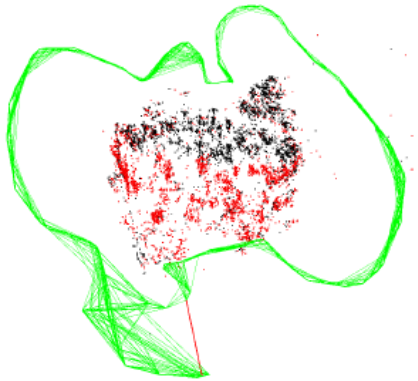
Loop Closing: Optimization



- Essential graph is spanning tree of covisibility graph (keeping strongest edges in tree structure) plus loop closure edges

- Pose graph optimization: Solve for pose of each camera that satisfies pairwise similarity transforms (edges in essential graph) as well as possible

- Update map points to be consistent with new poses



(d) Essential Graph

Break

https://www.youtube.com/watch?v=8DISRmsO2YQ

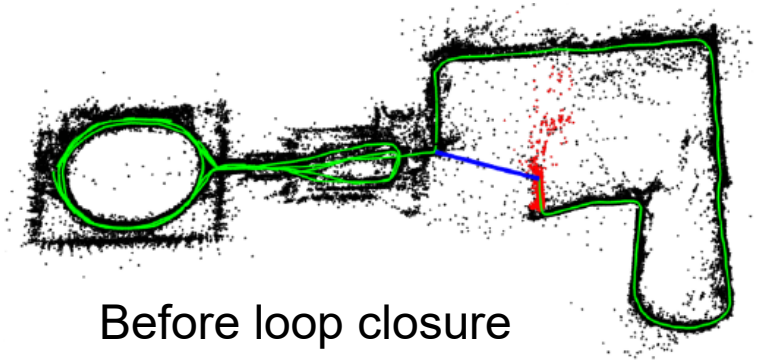
Results: NewCollege sequence

TABLE I
TRACKING AND MAPPING TIMES IN NEWCOLLEGE

Thread	Operation	Median (ms)	Mean (ms)	Std (ms)
TRACKING	ORB extraction	11.10	11.42	1.61
	Initial Pose Est.	3.38	3.45	0.99
	Track Local Map	14.84	16.01	9.98
	Total	30.57	31.60	10.39
LOCAL MAPPING	KeyFrame Insertion	10.29	11.88	5.03
	Map Point Culling	0.10	3.18	6.70
	Map Point Creation	66.79	72.96	31.48
	Local BA	296.08	360.41	171.11
	KeyFrame Culling	8.07	15.79	18.98
	Total	383.59	464.27	217.89

TABLE II
LOOP CLOSING TIMES IN NEWCOLLEGE

			Loop Detection (ms)		Loop Correction (s)		
Loop	KeyFrames	Essential Graph Edges	Candidates Detection	Similarity Transformation	Fusion	Essential Graph Optimization	Total (s)
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



Before loop closure

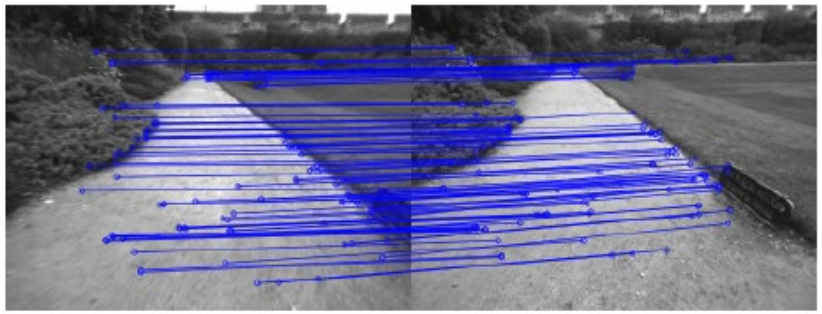
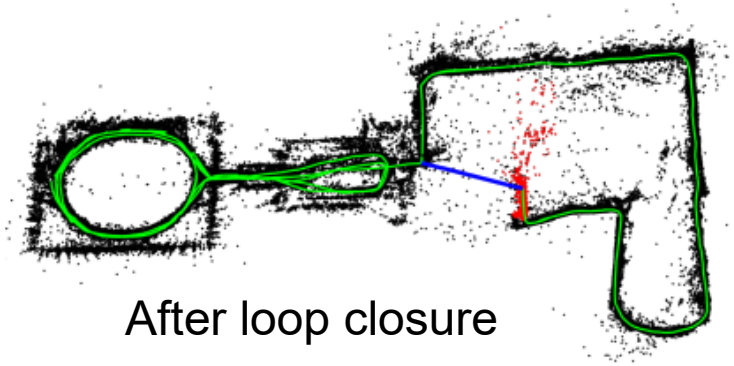


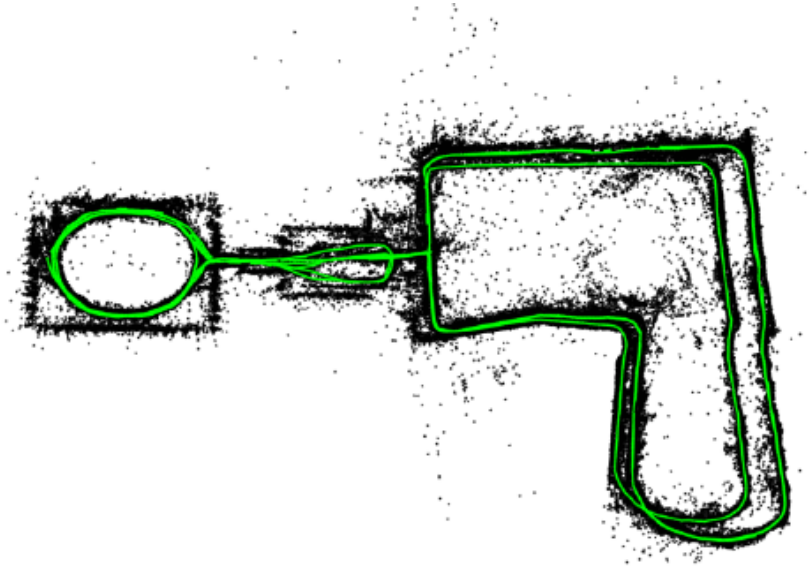
Fig. 4. Example of loop detected in the NewCollege sequence. We draw the inlier correspondences supporting the similarity transformation found.

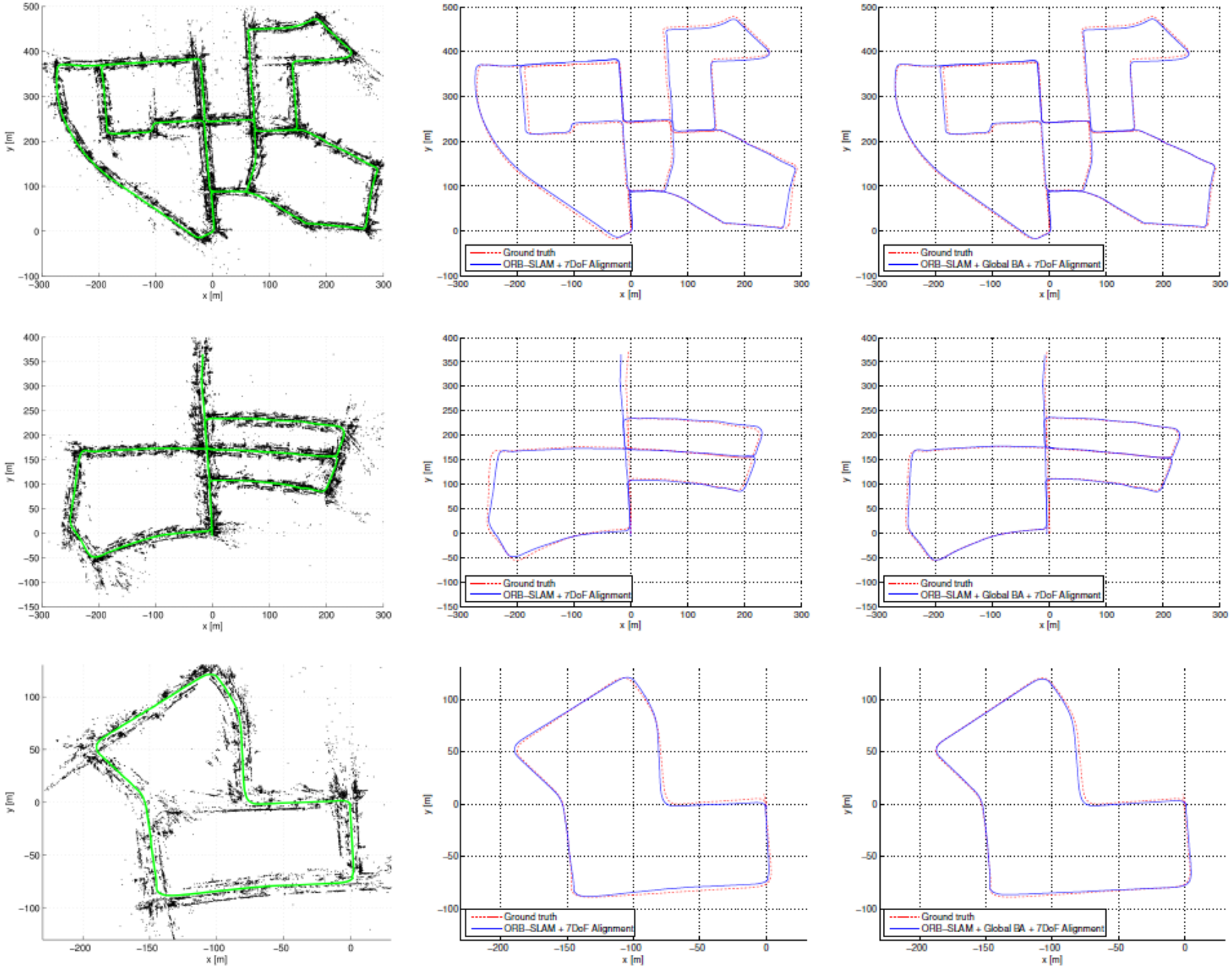


After loop closure

Results: NewCollege sequence

- Loop traveled in reverse direction is not matched and thus slightly misaligned





Global BA provides only slight improvement (because SLAM was already working well)

Localization accuracy

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	-

TABLE V
RESULTS OF OUR SYSTEM IN THE KITTI DATASET.

		ORB-SLAM		+ Global BA (20 its.)	
Sequence	Dimension (m×m)	KFs	RMSE (m)	RMSE (m)	Time BA (s)
KITTI 00	564 × 496	1391	6.68	5.33	24.83
KITTI 01	1157 × 1827	X	X	X	X
KITTI 02	599 × 946	1801	21.75	21.28	30.07
KITTI 03	471 × 199	250	1.59	1.51	4.88
KITTI 04	0.5 × 394	108	1.79	1.62	1.58
KITTI 05	479 × 426	820	8.23	4.85	15.20
KITTI 06	23 × 457	373	14.68	12.34	7.78
KITTI 07	191 × 209	351	3.36	2.26	6.28
KITTI 08	808 × 391	1473	46.58	46.68	25.60
KITTI 09	465 × 568	653	7.62	6.62	11.33
KITTI 10	671 × 177	411	8.68	8.80	7.64

ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras

Raúl Mur-Artal and Juan D. Tardós

2017

- Extension to stereo and RGBD

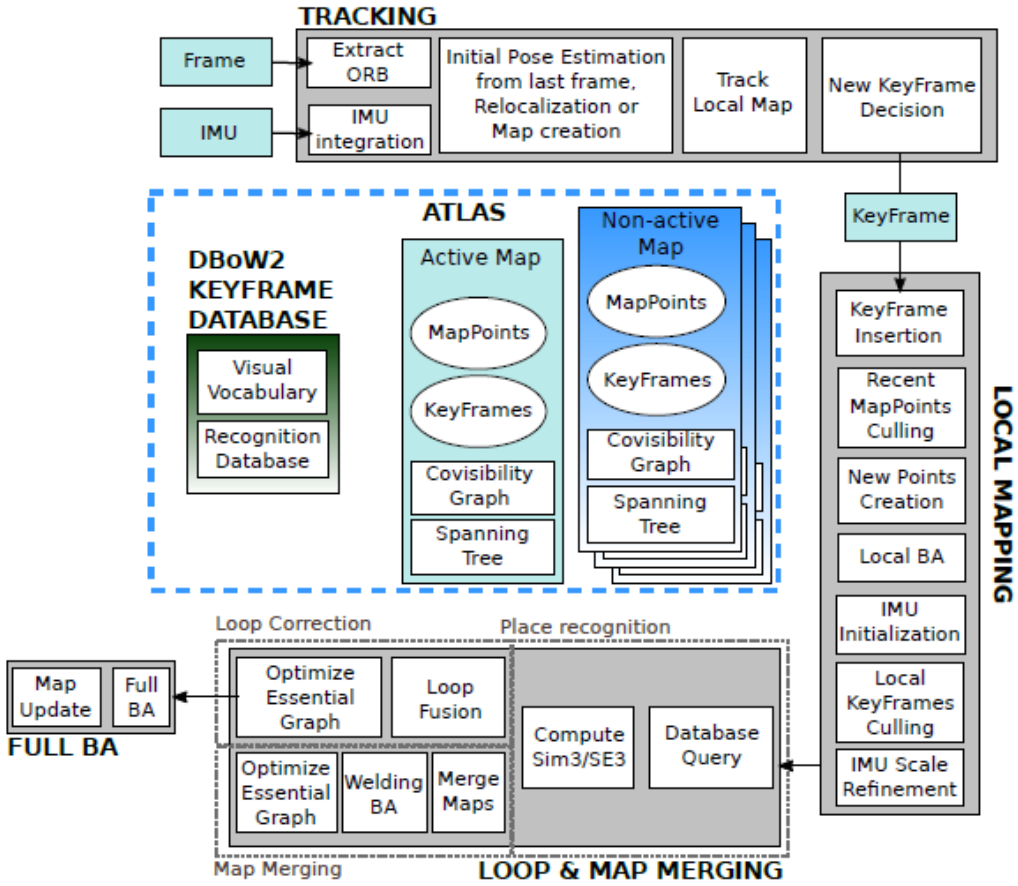
- Localization mode that localizes to map (without updating map) and uses frame-to-frame tracking when off map

ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM

Carlos Campos*, Richard Elvira*, Juan J. Gómez Rodríguez, José M.M. Montiel and Juan D. Tardós

2021

- Incorporates IMU (inertial measurement unit)
- Maintains several maps so can build over multiple sessions or start new map if tracking lost and merge later



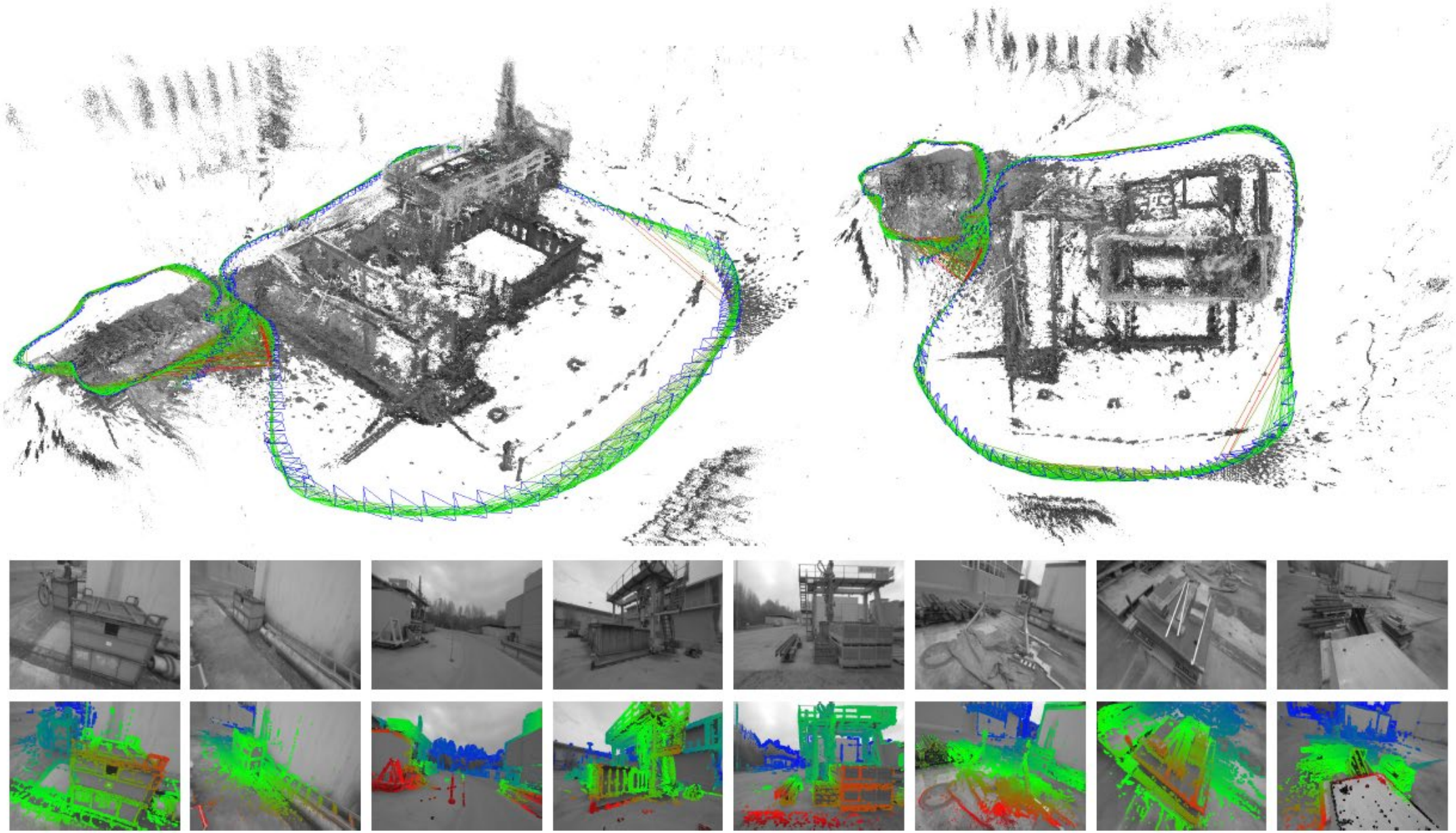
	SLAM or VO	Pixels used	Data association	Estimation	Relocali-zation	Loop closing	Multi Maps	Mono	Stereo	Mono IMU	Stereo IMU	Fisheye	Accuracy	Robustness	Open source
Mono-SLAM [13], [14]	SLAM	Shi Tomasi	Correlation	EKF	-	-	-	✓	-	-	-	-	Fair	Fair	[15][1]
PTAM [16]–[18]	SLAM	FAST	Pyramid SSD	BA	Thumbnail	-	-	✓	-	-	-	-	Very Good	Fair	[19]
LSD-SLAM [20], [21]	SLAM	Edgelets	Direct	PG	-	FABMAP PG	-	✓	✓	-	-	-	Good	Fair	[22]
SVO [23], [24]	VO	FAST+ Hi.grad.	Direct	Local BA	-	-	-	✓	✓	-	-	✓	Very Good	Very Good	[25][2]
ORB-SLAM2 [2], [3]	SLAM	ORB	Descriptor	Local BA	DBoW2	DBoW2 PG+BA	-	✓	✓	-	-	-	Exc.	Very Good	[26]
DSO [27]–[29]	VO	High grad.	Direct	Local BA	-	-	-	✓	✓	-	-	✓	Fair	Very Good	[30]
DSM [31]	SLAM	High grad.	Direct	Local BA	-	-	-	✓	-	-	-	-	Very Good	Very Good	[32]
MSCKF [33]–[36]	VO	Shi Tomasi	Cross correlation	EKF	-	-	-	✓	-	✓	✓	-	Fair	Very Good	[37][3]
OKVIS [38], [39]	VO	BRISK	Descriptor	Local BA	-	-	-	-	-	✓	✓	✓	Good	Very Good	[40]
ROVIO [41], [42]	VO	Shi Tomasi	Direct	EKF	-	-	-	-	-	✓	✓	✓	Good	Very Good	[43]
ORBSLAM-VI [4]	SLAM	ORB	Descriptor	Local BA	DBoW2	DBoW2 PG+BA	-	✓	-	✓	-	-	Very Good	Very Good	-
VINS-Fusion [7], [44]	VO	Shi Tomasi	KLT	Local BA	DBoW2	DBoW2 PG	✓	-	✓	✓	✓	✓	Good	Exc.	[45]
VI-DSO [46]	VO	High grad.	Direct	Local BA	-	-	-	-	-	✓	-	-	Very Good	Exc.	-
BASALT [47]	VO	FAST	KLT (LSSD)	Local BA	-	ORB BA	-	-	-	-	✓	✓	Very Good	Exc.	[48]
Kimera [8]	VO	Shi Tomasi	KLT	Local BA	-	DBoW2 PG	-	-	-	-	✓	-	Good	Exc.	[49]
ORB-SLAM3 (ours)	SLAM	ORB	Descriptor	Local BA	DBoW2	DBoW2 PG+BA	✓	✓	✓	✓	✓	✓	Exc.	Exc.	[5]

			MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203	Avg[1]
Monocular	ORB-SLAM [4]	ATE[2,3]	0.071	0.067	0.071	0.082	**0.060**	**0.015**	0.020	-	**0.021**	**0.018**	-	0.047*
	DSO [27]	ATE	0.046	0.046	0.172	3.810	0.110	0.089	0.107	0.903	0.044	0.132	1.152	0.601
	SVO [24]	ATE	0.100	0.120	0.410	0.430	0.300	0.070	0.210	-	0.110	0.110	1.080	0.294*
	DSM [31]	ATE	0.039	0.036	0.055	**0.057**	0.067	0.095	0.059	0.076	0.056	0.057	**0.784**	**0.126**
	ORB-SLAM3 (ours)	ATE	**0.016**	**0.027**	**0.028**	0.138	0.072	0.033	**0.015**	**0.033**	0.023	0.029	-	0.041*
Stereo	ORB-SLAM2 [3]	ATE	0.035	**0.018**	0.028	0.119	0.060	**0.035**	**0.020**	**0.048**	**0.037**	0.035	-	0.044*
	VINS-Fusion [44]	ATE	0.540	0.460	0.330	0.780	0.500	0.550	0.230	-	0.230	0.200	-	0.424*
	SVO [24]	ATE	0.040	0.070	0.270	0.170	0.120	0.040	0.040	0.070	0.050	0.090	0.790	0.159
	ORB-SLAM3 (ours)	ATE	**0.029**	0.019	**0.024**	**0.085**	**0.052**	**0.035**	0.025	0.061	0.041	**0.028**	**0.521**	**0.084**
Monocular Inertial	MCSKF [33]	ATE[5]	0.420	0.450	0.230	0.370	0.480	0.340	0.200	0.670	0.100	0.160	1.130	0.414
	OKVIS [39]	ATE[5]	0.160	0.220	0.240	0.340	0.470	0.090	0.200	0.240	0.130	0.160	0.290	0.231
	ROVIO [42]	ATE[5]	0.210	0.250	0.250	0.490	0.520	0.100	0.100	0.140	0.120	0.140	0.140	0.224
	ORBSLAM-VI [4]	ATE[2,3] scale error[2,3]	0.075 0.5	0.084 0.8	0.087 1.5	0.217 3.5	0.082 0.5	**0.027** 0.9	0.028 0.8	- -	**0.032** 0.2	0.041 1.4	0.074 0.7	0.075* 1.1*
	VINS-Mono [7]	ATE[4]	0.084	0.105	0.074	0.122	0.147	0.047	0.066	0.180	0.056	0.090	0.244	0.110
	VI-DSO [46]	ATE scale error	**0.062** 1.1	0.044 0.5	0.117 0.4	0.132 0.2	0.121 0.8	0.059 1.1	0.067 1.1	0.096 0.8	0.040 1.2	0.062 0.3	0.174 0.4	0.089 0.7
	ORB-SLAM3 (ours)	ATE scale error	**0.062** 1.4	**0.037** 0.3	**0.046** 0.8	**0.075** 0.5	**0.057** 0.3	0.049 2.0	**0.015** 0.6	**0.037** 2.2	0.042 0.7	**0.021** 0.4	**0.027** 1.0	**0.043** 0.9
Stereo Inertial	VINS-Fusion [44]	ATE[4]	0.166	0.152	0.125	0.280	0.284	0.076	0.069	0.114	0.066	0.091	0.096	0.138
	BASALT [47]	ATE[3]	0.080	0.060	0.050	0.100	**0.080**	0.040	0.020	0.030	**0.030**	0.020	-	0.051*
	Kimera [8]	ATE	0.080	0.090	0.110	0.150	0.240	0.050	0.110	0.120	0.070	0.100	0.190	0.119
	ORB-SLAM3 (ours)	ATE scale error	**0.036** 0.6	**0.033** 0.2	**0.035** 0.6	**0.051** 0.2	0.082 0.9	**0.038** 0.8	**0.014** 0.6	**0.024** 0.8	0.032 1.1	**0.014** 0.2	**0.024** 0.2	**0.035** 0.6

Other notes from ORB-SLAM 3 conclusions

- Main failure of ORB-SLAM is low-texture environments
- “Without question, stereo-inertial SLAM provides the most accurate solution”
- Monocular-inertial SLAM almost as good but can break if the camera purely rotates
- IMU sensors can be difficult to initialize in slow or steady motion

LSD-SLAM [Engel, Schops, Cremers, ECCV 2014]



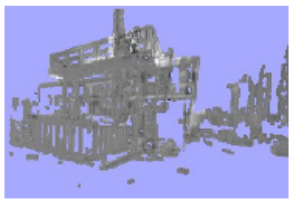
LSD-SLAM: Overview (Large-Scale Direct SLAM)

- “Direct” visual odometry
  - Matching/alignment uses pixel correlation instead of features
- Dense matches in textured/edge regions
- Alignment between keyframes estimates similarity transform to account for scale drift
- Depth and uncertainty estimated per pixel

Tracking

New Image
(640 x 480 at 30Hz)

Track on Current KF:
→ estimate SE(3) transformation



$$\min_{\xi \in \mathfrak{se}(3)} \sum_{\mathbf{p}} \left\| \frac{r_p^2(\mathbf{p},\boldsymbol{\xi})}{\sigma^2_{r_p(\mathbf{p},\boldsymbol{\xi})}} \right\|_\delta$$

tracking reference

(See Sec. 3.3)

Depth Map Estimation

Take KF?

yes

no

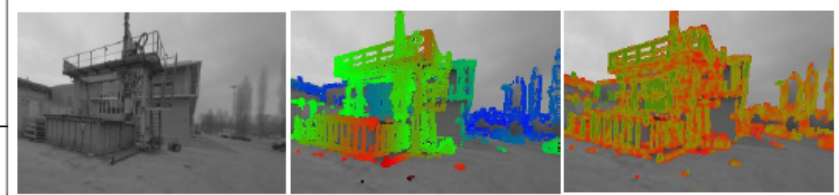
Create New KF
→ propagate depth map to new frame
→ regularize depth map

Refine Current KF
→ small-baseline stereo
→ probabilistically merge into KF
→ regularize depth map

replace KF

refine KF

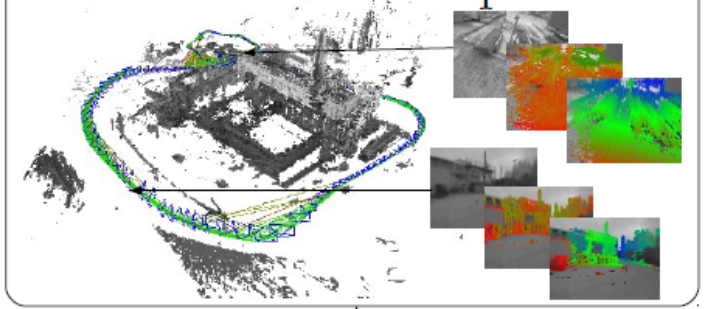
Current KF



(See Sec. 3.4)

Map Optimization

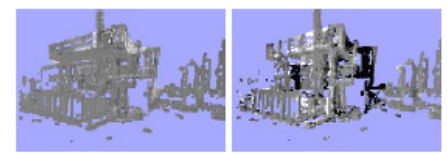
Current Map



add to map

Add KF to Map
→ find closest keyframes
→ estimate Sim(3) edges

$$\min_{\xi \in \mathfrak{sim}(3)} \sum_{\mathbf{p}} \left\| \frac{r_p^2(\mathbf{p},\boldsymbol{\xi})}{\sigma^2_{r_p(\mathbf{p},\boldsymbol{\xi})}} + \frac{r_d^2(\mathbf{p},\boldsymbol{\xi})}{\sigma^2_{r_d(\mathbf{p},\boldsymbol{\xi})}} \right\|_\delta$$



(See Sec. 3.2, 3.5 and 3.6)

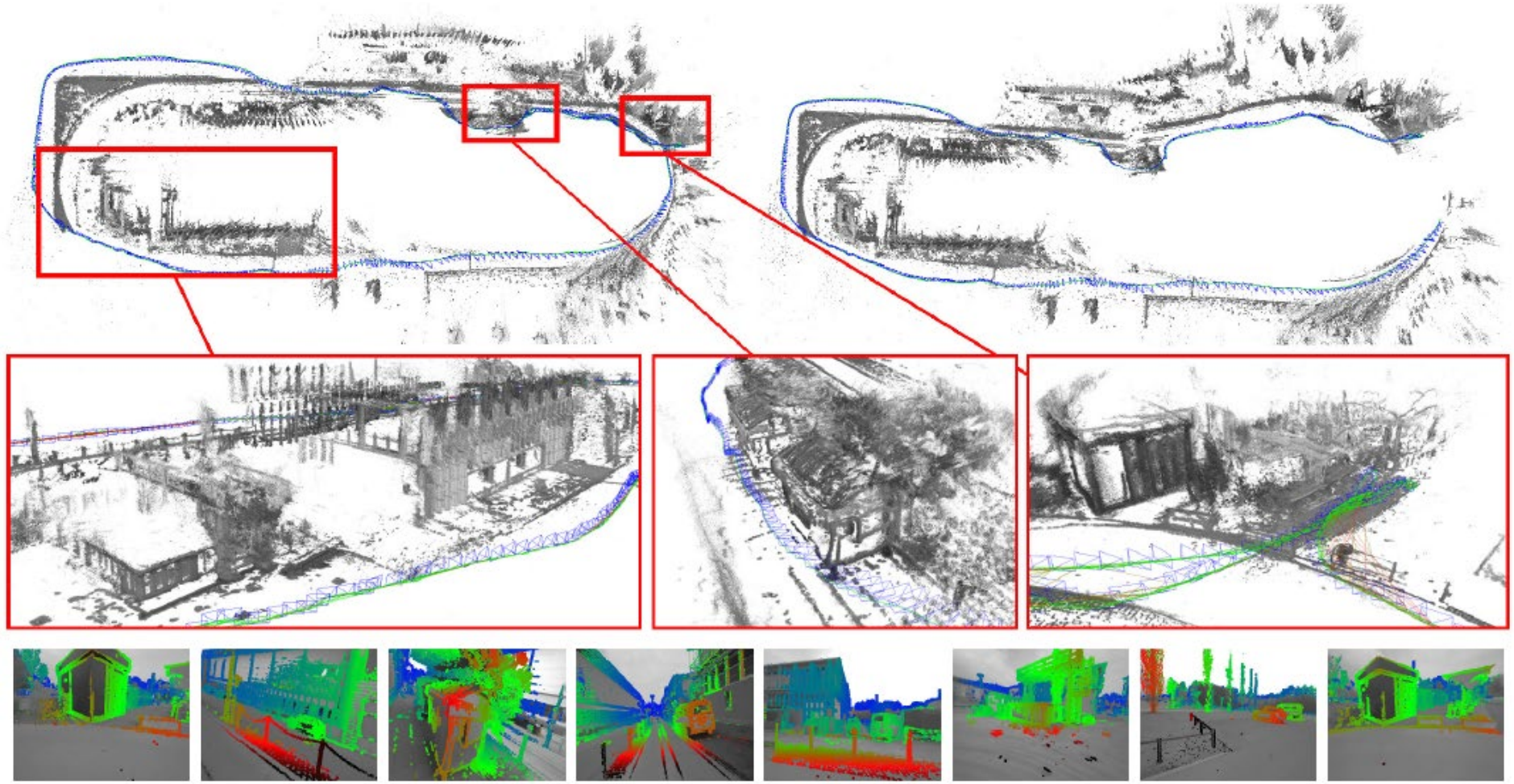


Fig. 7: Loop closure for a long and challenging outdoor trajectory (after the loop closure on the left, before on the right). Also shown are three selected close-ups of the generated pointcloud, and semi-dense depth maps for selected keyframes.

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	-

Open problems / research ideas

- Very large scale mapping and relocalization
  - Maintain maps of an entire campus or city while keeping to a memory budget
- Mix of features types
  - KLT for short-range tracking
  - ORB/SIFT for medium-range mapping
  - Deep features for loop closure and relocalization
- Incorporate single-view depth prediction or completion(?)

Summary

- SLAM uses a combination of incremental (frame-to-frame tracking, keyframe addition) and global (pose graph optimization) techniques
- SLAM usually aims to be real-time on CPU and principal aim is camera localization rather than scene reconstruction
- Compared to SfM methods SLAM methods tend to be robust but less precise, i.e. does not get totally lost but not accurate enough localization for good MVS