Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping

Introduce

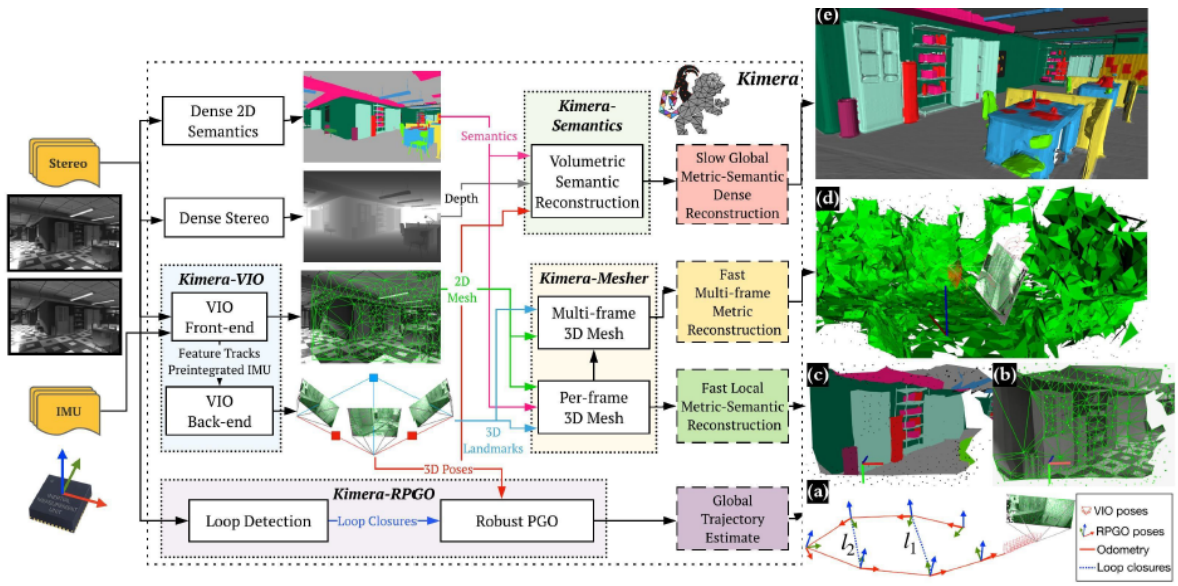
This is an open-source C++ library for real-time metric-semantic visual-inertial SLAM.

designed with modularity and has four key components:

- Kimera-VIO: fast and accurate IMU-rate state estimation
- Kimera-RPGO: global pose graph optimization
- Kimera-Mesher: computes a fast per-frame and multi-frame regularized 3D mesh to support obstacle avoidance.
- Kimera-Semantics: builds a slower but more accurate global 3D mesh using a volumetric approach

Framework

pipeline



VIO

- front-end

  use GFTT and LK find stereo match

  use RANSAC(mono 5p, stereo 3p, mono+IMU 2p, stereo+IMU 1p)

  Feature detection, stereo matching and geometric verification are executed at each keyframe

- back-end

  at keyframe preintegrated IMU and visual measurements are added to factor graph(iSAM2)

  estimates the 3D position of the observed features using DLT

Graph optimize

- loop closure

  loop closure detection relies on the DBoW2 library

  reject outlier and PGO

- pose graph optimization(PCM)

  adds an odometry consistency check on the loop closures

  incrementally updates the set of consistent measurements

  pose-graph

Mesh Reconstruction

- per-frame

  perform a 2D Delaunay triangulation over tracked 2D features

  back-project the 2D Delaunay triangulation to generate a 3D mesh

  option to semantically label the resulting mesh,

- multi-frame

  loop over exist vertices and triplets and add vertices and triplets that are in the per-frame mesh but are missing in the multi-frame one.

  outlier the mesh out of local map

Semantic

- global mesh

  keyframe find dense pcl

  use bundled raycasting find a voxel-based map(TSDF)

- semantic

Result

- ATE

	RMSE ATE [cm]						
Seq.	OKVIS	MSCKF	ROVIO	VINS-Mono	Kimera-VIO	VINS-Mono-LC	Kimera-RPGO
MH_01	16	42	21	15	**11**	12	**8**
MH_02	22	45	25	15	**10**	12	**9**
MH_03	24	23	25	22	**16**	13	**11**
MH_04	34	37	49	32	**24**	18	**15**
MH_05	47	48	52	**30**	35	**21**	24
V1_01	9	34	10	8	**5**	7	**5**
V1_02	20	20	10	11	**8**	**8**	11
V1_03	24	67	14	18	**7**	19	**12**
V2_01	13	10	12	**8**	**8**	8	**7**
V2_02	16	16	14	16	**10**	16	**10**
V2_03	29	113	**14**	27	21	22	**19**

- overall