

SLAM Benchmarking

Introduction

This is a repository of state of the art simultaneous localization and mapping (SLAM) approaches for the purpose of benchmarking. Requires ROS installation, among other various third party libraries which are explained in more detail in each source's wiki.

ROS installation can be found here: <http://wiki.ros.org/kinetic/Installation>

As well as a brief tutorial explaining the repository structure:

<http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment>

Packages that require ROS are located in the `$CATKIN_WS/src` directory and are also denoted here with an asterisk (*).

Note: After each new package added and built in the `$CATKIN_WS/src` directory, be sure to run commands:

```
source /opt/ros/kinetic/setup.bash
source $CATKIN_WS/devel/setup.bash
```

these locate and initialize new packages for ROS for building and launching.

Compatible, standard datasets:

EuRoC:

<https://projects.asl.ethz.ch/datasets/doku.php?id=k mavvisualinertialdatasets#downloads>

KITTI (in progress):

http://www.cvlibs.net/datasets/kitti/raw_data.php?type=calibration

Environment setup:

In `~/ .bashrc` add lines to end of file (assuming having installed ROS):

```
source /opt/ros/kinetic/setup.bash
export SLAM_BM=<path to repo root>
```

Also be sure to create an output directory in your `SLAM_BM` directory:

```
cd $SLAM_BM
mkdir output
```

This will be used as the dump for pose logging and file parsing.

Also, for building ROS SLAM code, where your `gflags` is, add the following line to bottom to enable shared library linking (or, alternatively, checkout MARS branch `slam_benchmarking`):

```
set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -fPIC") # for linking with shared
object libraries
```

Third Party Libraries and Installation Instructions

kindr (rovio): <https://github.com/ANYbotics/kindr>

lightweight_filtering (rovio): Included with ROVIO

sophus (svo): <https://github.com/strasdat/Sophus>

fast (svo): <https://github.com/uzh-rpg/fast>

g2o (svo): <https://github.com/RainerKuemmerle/g2o>

rpg_vikit (svo): https://github.com/uzh-rpg/rpg_vikit

ceres-solver (vins-mono): <http://ceres-solver.org/installation.html>

Build commands:

```
cd $REPO_ROOT
python3 scripts/install_third_party.py
```

Please refer to the library's source for any troubleshooting. Other system libraries may be needed.

MARS MSCKF

Source: <https://github.com/MARSLab-UMN/MarsFramework>

Paper:

OKVIS

Source: <https://github.com/ethz-asl/okvis>

Paper: <http://www.roboticsproceedings.org/rss09/p37.pdf>

Installation guide:

A thorough guide of the necessary system libraries and third party packages, including those listed above, can be found in the wiki, which can be found in the source link above. Make sure to double-check the necessary system libraries are installed for each required third-party package and library package.

Build commands:

```
cd $REPO_ROOT
python3 scripts/build_slam_code.py okvis
```

Run code:

```
./okvis_app_synchronous <path to yaml config (e.g.
okvis/config/config_fpga_p2_euroc.yaml)> <path to dataset (e.g. mav0)>
```

ROVIO*

Source: <https://github.com/ethz-asl/rovio>

Paper: <https://arxiv.org/pdf/1712.00036.pdf>

Installation guide:

A thorough guide of the necessary system libraries and third party packages, including those listed above, can be found in the wiki, which can be found in the source link above. Make sure to double-check the necessary system libraries are installed for each required third-party package and library package.

Build command:

```
cd $ROOT_REPO
python3 scripts/build_slam_code.py rovio
```

Run code:

Configure `$CATKIN_WS/src/rovio/launch/rovio_rosbag_node.launch` file to desired dataset location

```
roslaunch rovio rovio_rosbag_node.launch
```

SVO

Source: https://github.com/uzh-rpg/rpg_svo

Paper: http://rpg.ifi.uzh.ch/docs/ICRA14_Forster.pdf

Installation guide:

A thorough guide of the necessary system libraries and third party packages, including those listed above, can be found in the wiki, which can be found in the source link above. Make sure to double-check the necessary system libraries are installed for each required third-party package and library package.

Build command:

```
cd $REPO_ROOT
python3 scripts/build_slam_code.py svo
```

Run code:

```
cd $REPO_ROOT/svo/rpg_svo/svo/bin
./test_pipeline
```

Vins-Mono*

Source: <https://github.com/HKUST-Aerial-Robotics/VINS-Mono>

Paper: <https://arxiv.org/pdf/1708.03852.pdf>

System Requirements: Ubuntu 16.04

Installation guide:

A thorough guide of the necessary system libraries and third party packages, including those listed above, can be found in the wiki, which can be found in the source link above. Make sure to double-check the necessary system libraries are installed for each required third-party package and library package.

Build commands:

```
cd $REPO_ROOT
python3 scripts/build_slam_code.py vins-mono
```

Run code:

Open three new terminals.

In the first, run `roslaunch vins_estimator euroc.launch`

In the second, run `roslaunch vins_estimator vins_rviz.launch`

Finally, in the third, run `rosbag play YOUR_PATH_TO_DATASET/MH_01_easy.bag`

ORB Slam

Source: https://github.com/raulmur/ORB_SLAM2

Paper: <http://webdiis.unizar.es/~raulmur/MurMontielTardosTRO15.pdf>

Build commands:

```
cd $REPO_ROOT
python3 scripts/build_slam_code.py orb-slam
```

Run code (all one command):

```
./Examples/Monocular/mono_euroc Vocabulary/ORBvoc.txt
```

```
Examples/Monocular/EuRoC.yaml <path to euroc dataset (e.g. mav0)>/cam0/data
Examples/Monocular/EuRoC_TimeStamps/<EuRoC sequence (e.g. MH01.txt)>.txt
```

Further Notes

*Build All ROS packages *third party libraries must be built first**

```
cd $CATKIN_WS
catkin_make
```