SLAM

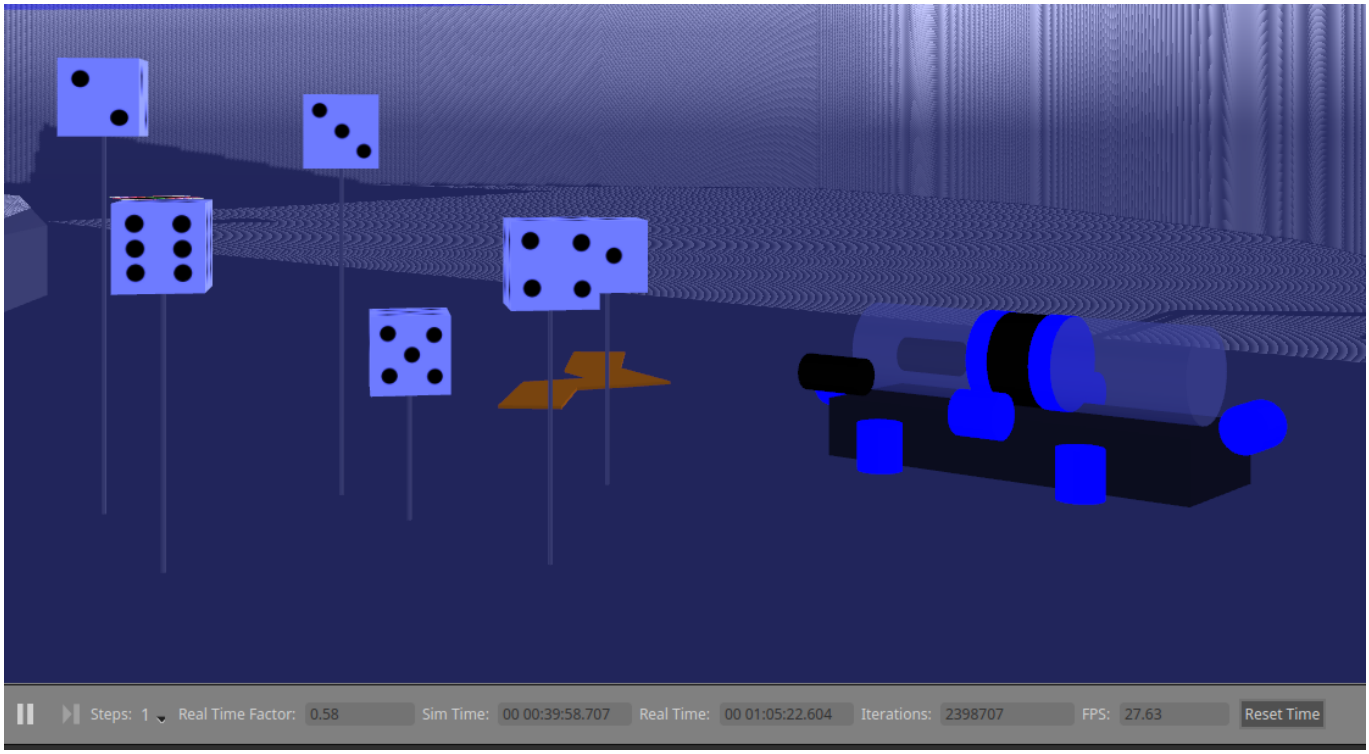
Simultaneous Localization and Mapping

Overview

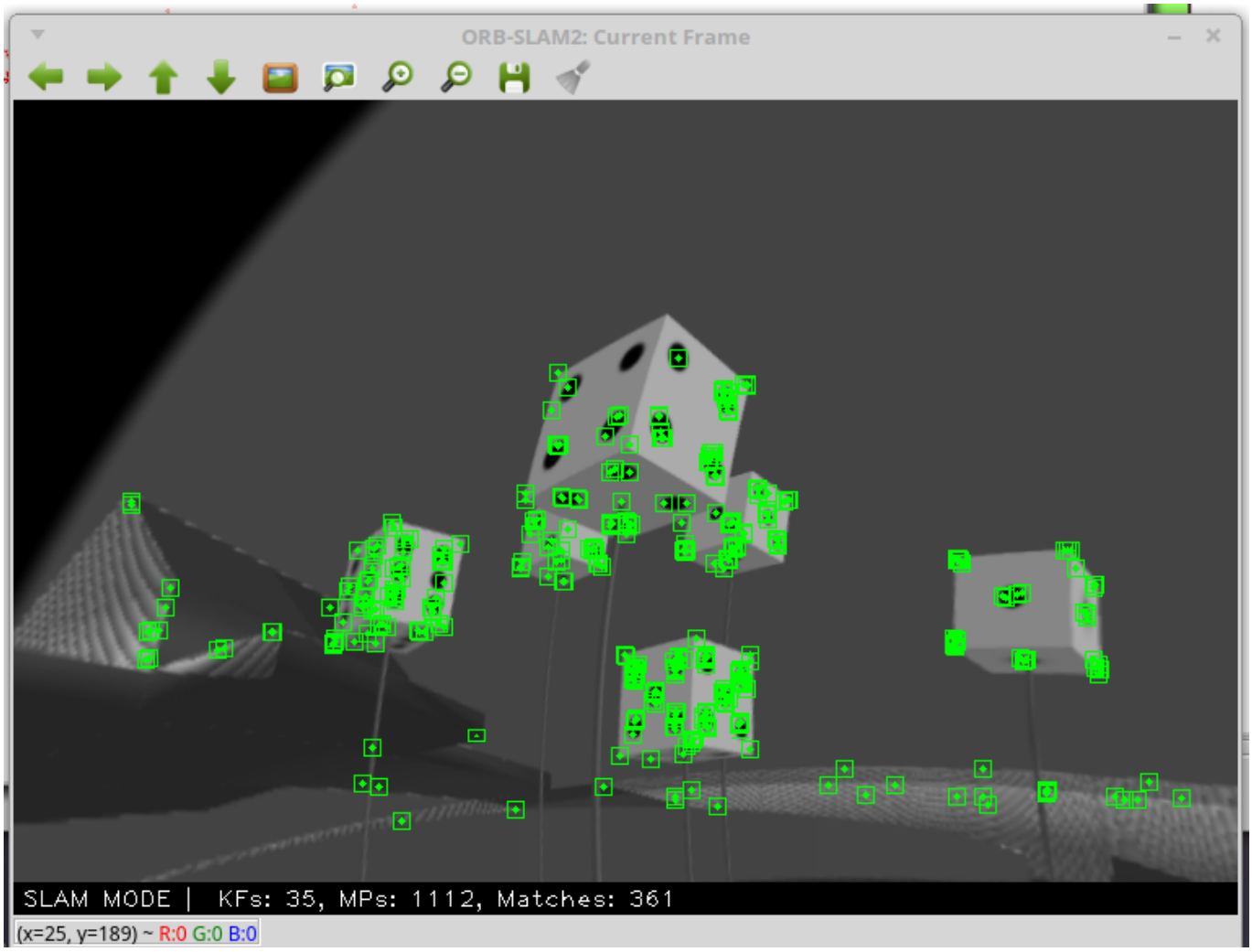
In robot navigation, a SLAM algorithm is used to construct a map of the robot's environment, while simultaneously locating the robot within that map. There are many different SLAM algorithms, but we are currently using a visual based system using the sub's right and left cameras. This allows us to link the system to Object Detection.

The specific system we are using is ORB-SLAM2, an open source feature based visual slam system which we modified for the sub.

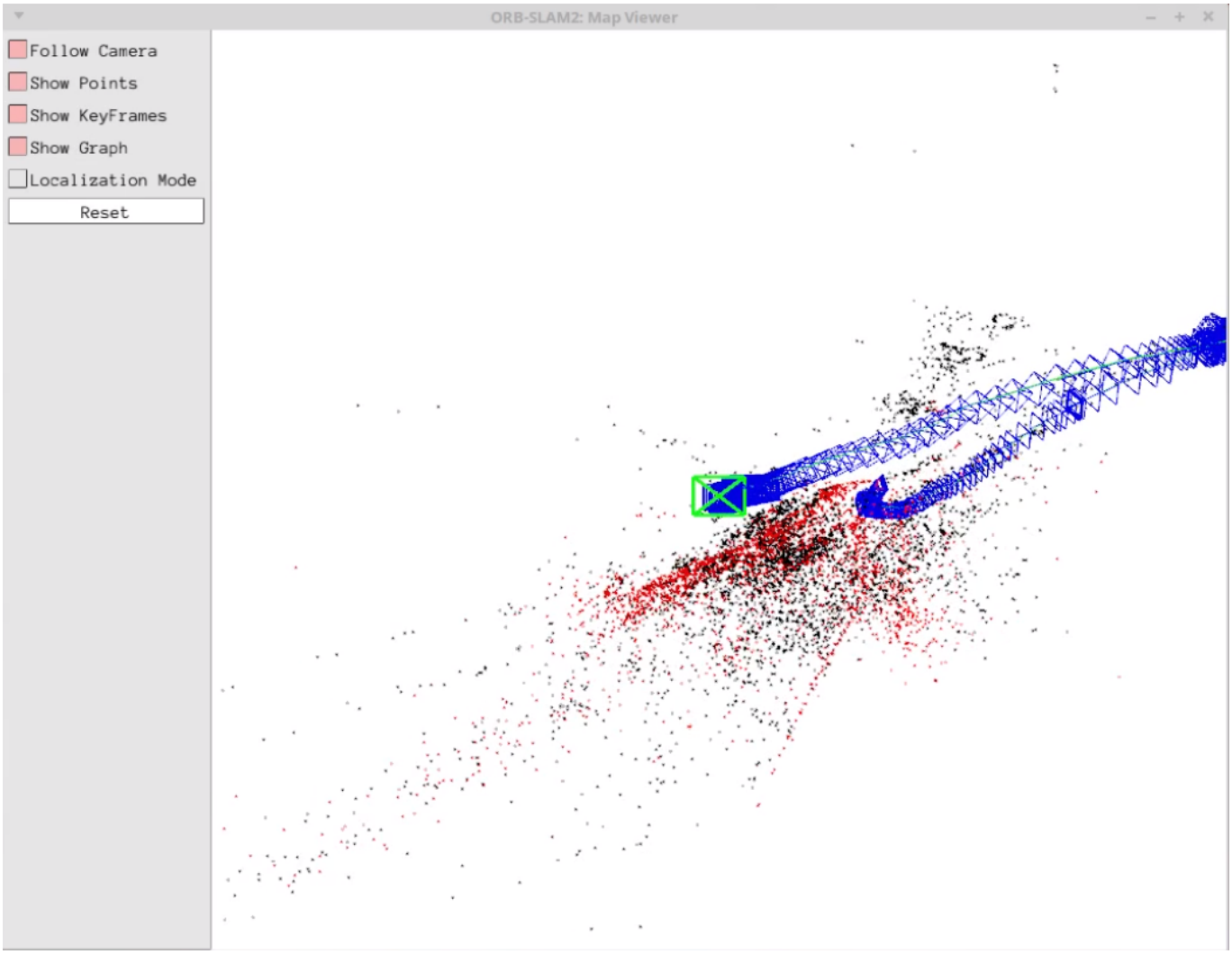
The algorithm works by detecting features (such as edges and corners) in an image, and locates them in space using triangulation with other known map points.



The sub in simulated environment.



The view of a single keyframe with detected map points.



A viewer that plots the detected map points.

Structure

The SLAM algorithm is complex, but it links to the rest of the sub's system through a single node at ~/ros/src/robosub_orb_slam/src/ros_stereo.cc. The node:

Subscribes to:

- /camera/left/image_raw - collects image data from left robosub camera.
- /camera/right/image_raw - collects image data from right robosub camera.

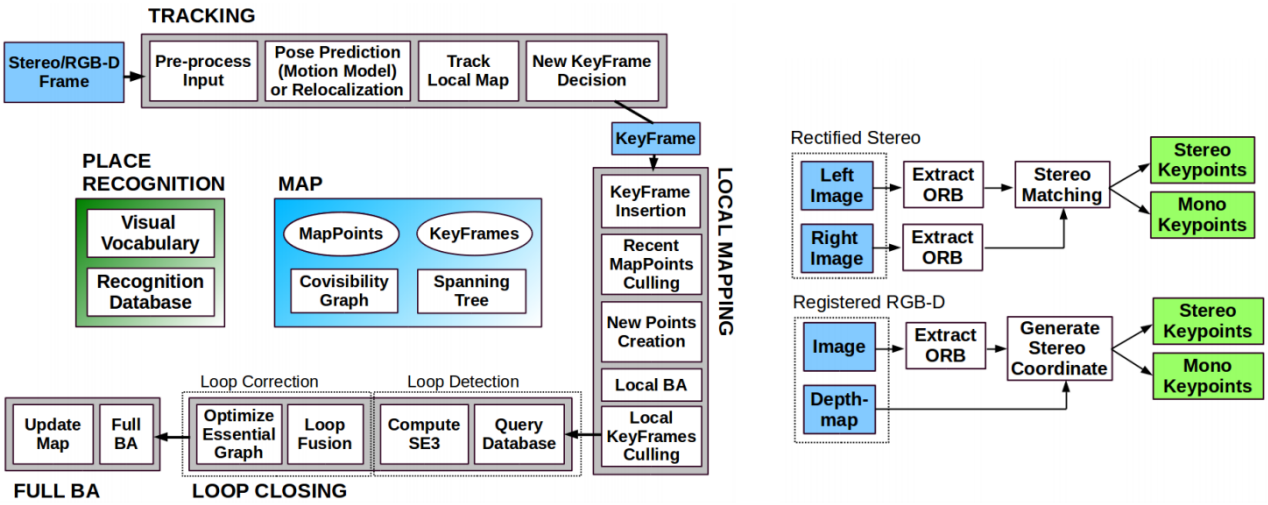
Publishes to:

- /SLAMpoints - the 3d location of the map points in space, and the 2d location of the map points on each image frame.

How to Run

```
$ roslaunch robosub_orb_slam slam.launch use_viewer:=True
```

How it Works



(a) System Threads and Modules. (b) Input pre-processing

ORB-SLAM2 is a feature based algorithm that takes keyframes from video output and extracts keypoints or features (such as corners), and uses them to establish location of the sub and its surroundings.

It consists of three main modules: Tracking, Local Mapping, and Loop Closing.

Tracking

-Tracking localizes the camera by comparing features in a local map.

-Detects features using the FAST Algorithm.

-Describes features using ORB Algorithm.

-Selects a new keyframe.

-If localization is lost, uses Place Recognition module to relocate.

The tracking part localizes the camera and decides when to insert a new keyframe. Features are matched with the previous frame and the pose is optimized using motion-only bundle adjustment. The features extracted are FAST corners. (for res. till 752×480, 1000 corners should be good, for higher (KITTI 1241×376) 2000 corners works). Multiple scale-levels (factor 1.2) are used and each level is divided into a grid in which 5 corners per cell are attempted to be extracted. These FAST corners are then described using ORB. The initial pose is estimated using a constant velocity motion model. If the tracking is lost, the place recognition module kicks in and tries to re-localize itself. When there is an estimation of the pose and feature matches, the co-visibility graph of keyframes, that is maintained by the system, is used to get a local visible map. This local map consists of keyframes that share map point with the current frame, the neighbors of these keyframes and a reference keyframe which share the most map points with the current frame. Through re-projection, matches of the local map are searched on the frame and the camera pose is optimized using these matches. Finally is decided if a new Keyframe needs to be created, new keyframes are inserted very frequently to make tracking more robust. A new keyframe is created when at least 20 frames has passed from the last keyframe,

and last global re-localization, the frame tracks at least 50 points of which less then 90% are point from the reference keyframe.

Local Mapping

-Keyframes are added to co-visibility graph Spanning Tree.

-New Map points are creates by triangulating matching ORB features from different keyframes.

-Validity of map point is checked by seeing if it is found in other keyframes where it is predicted to be. Must be seen by at least 3 other keyframes.

Loop Closing

-Loop closing is when the sub recognizes that it has returned to a previous location and adjust map points to accommodate.

-To detect possible loops, check bag of words vectors in Place Recognition module of the current keyframe and its neighbors in the co-visibility graph.

-If loop candidate is found preform similarity transform.

-Fuse map points and preform bundle adjustment.

Map

Each map point stores:

- Its 3D position in the world coordinate system.
- ORB descriptor.
- The maximum dmax and minimum dmin distances at which the point can be observed, according to the scale invariance limits of the ORB features.

Place Recognition

From:
https://robosub.eecs.wsu.edu/wiki/ - **Palouse RoboSub Technical Documentation**

Permanent link:
https://robosub.eecs.wsu.edu/wiki/cs/slam/start?rev=1566169100

Last update: **2019/08/18 15:58**