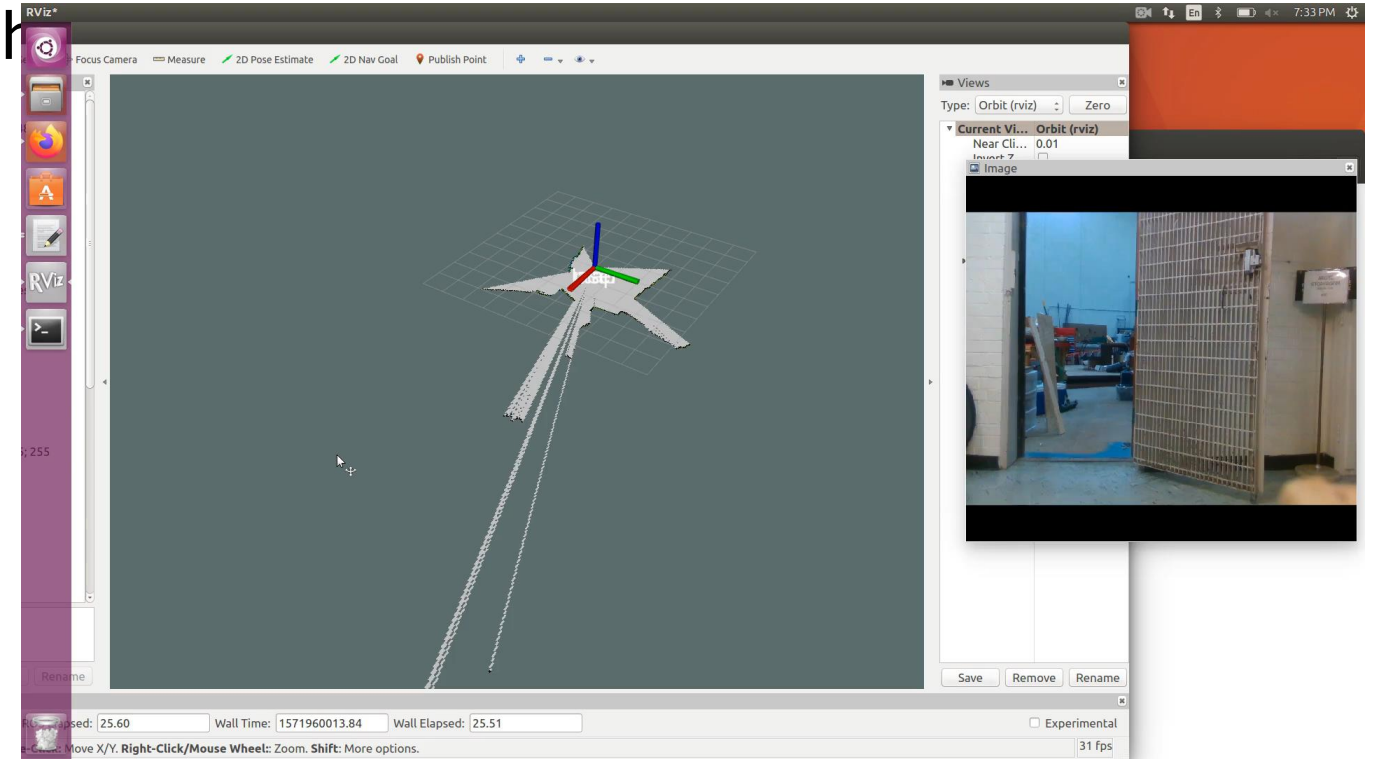
# SLAM & Perception

Dr. Pawandeep Singh Matharu

# What is SLAM

- Simultaneous Localization and Mapping:
  - Computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location with



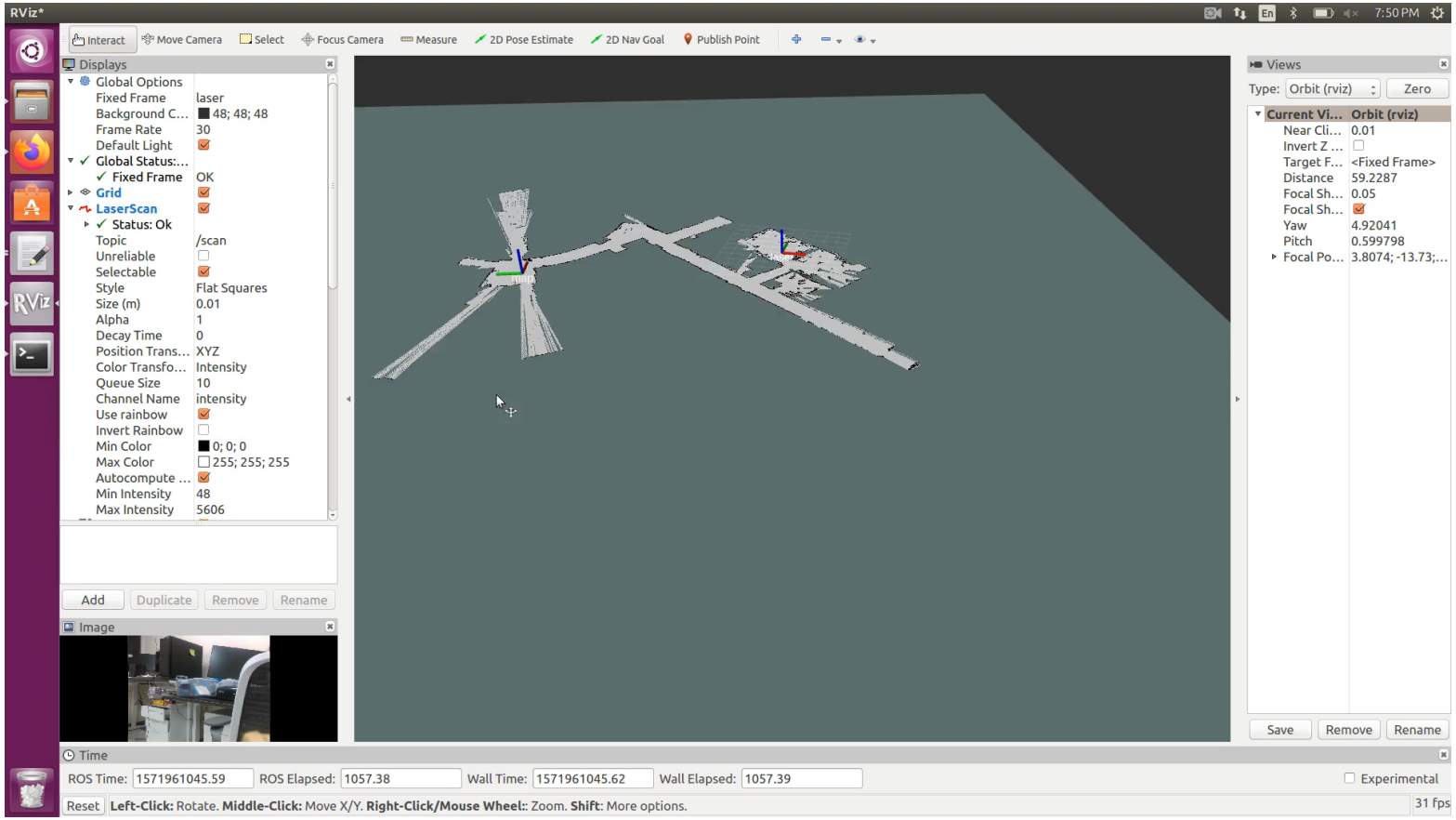
# Motivation for SLAM

- There are GPS-denied environments
  - Indoor
  - Underground
  - Underwater
  - Outdoor with tall obstacles (urban, forest, etc)
- Relative position within a local unknown environment is more important.

# Map

- A map defines a local reference frame.
- Maps allow robots to efficiently carry out their tasks, such as localization, path planning.
- 2D occupancy grid map is used in ROS community
  - Describe the environment as a grid of cells.
  - Each cell holds a probability value that the cell is occupied or not ([0, 1], 0 represents open space)
  - -1 or 0.5 indicates unexplored space (space beyond obstacles).

# 2D Map

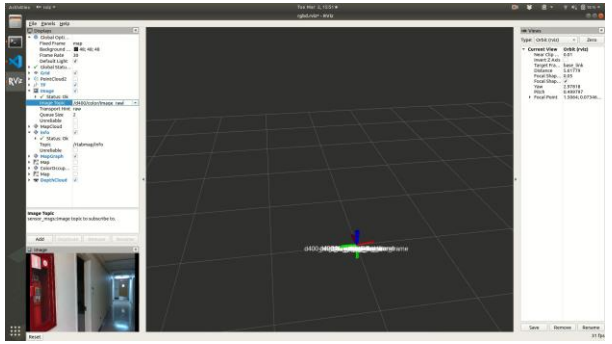



Gray: Unexplored

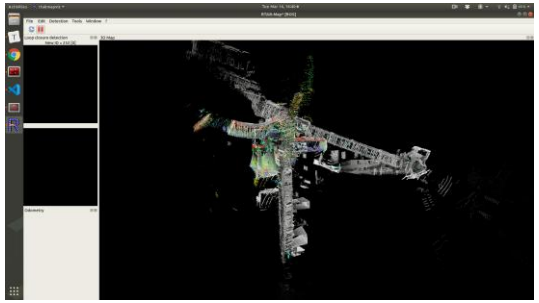
White: Unoccupied

Black: Occupied

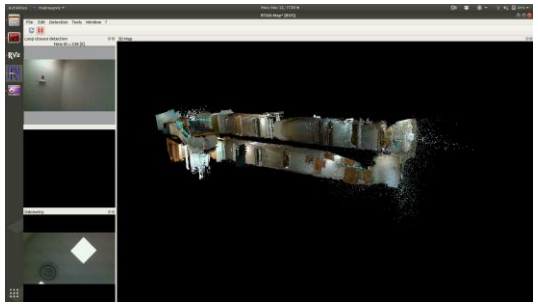
# 3D Map



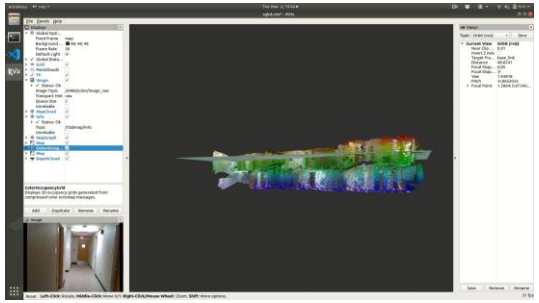
Mapping of two floors (1st and mezzanine floors)



L515



L515 + T265



D435i + T265

# Localization

Task for localization: pose of robot

- Position
- Orientation

Methods for Localization:

- **GPS**:
  - cannot tell orientation
  - positional error is also large
  - global map required
- **Dead reckoning**: Integration of acceleration and velocity from inertial measurement sensor
  - Accumulated error
- **SLAM**:
  - Large calculation burden

# SLAM

- Computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it.
- A chicken or egg problem:
  - An unbiased map is needed for localization.
  - An accurate pose estimate is needed to build the map.
- Odometry information:
  - Control command
  - IMU readings
  - Iterative closest point (ICP) to match current measurement with respect to the current map.
  - Visual odometry
  - Kalman Filter/Particle Filter to match the predicted measurement with the actual measurements.

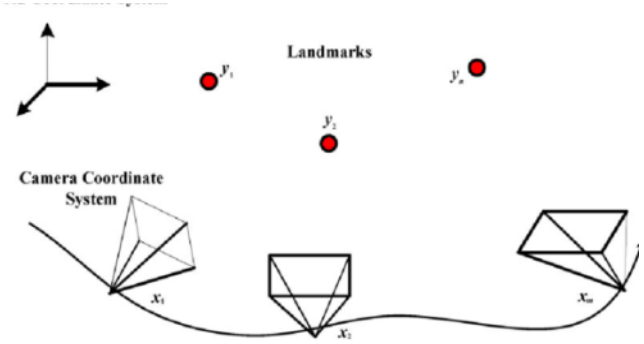
# Sensors for SLAM

**Sensors for mapping**

- LiDAR (Ouster)
- Depth camera (Realsense D435i camera)
- General RGB camera for Visual SLAM

**Sensors for localization**

- Visual odometry (Realsense T265 camera)
- Inertial Measurement Units (IMUs)
- Encoders



# SLAM Techniques

- **Scan-based SLAM**: Directly utilizes unprocessed scanned data as observations which are generally sets of points obtained from range sensors such as a sonar and a laser range finder.
- **Feature-based SLAM**: Extracts a set of features including different types of geometric models such as points, lines, curvatures, and any arbitrary shapes from the observation and uses as landmarks.

# tf

Many 3D coordinate frames will be involved for even a simple task for your robot.

- Base frame for the vehicle
- World frame for the map
- Laser_scan frame for the localization sensor
- Camera frame for the visual camera
- Gripper/arm/head/another_robot frame, etc.

tf is used in ros to track all these frames over time.

# Useful Links

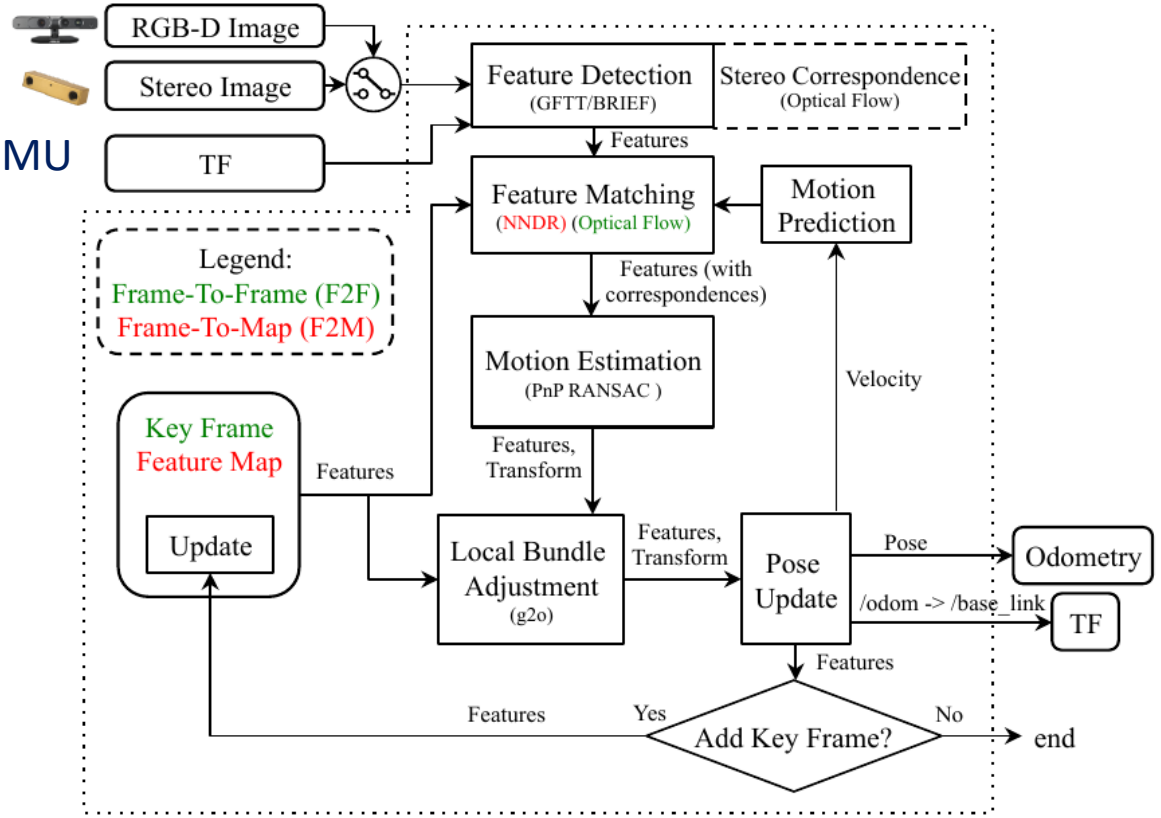
- SLAM simulation
  - https://google-cartographer-ros.readthedocs.io/en/latest/
- 2D SLAM
  - Gmapping (Needs odometry): http://wiki.ros.org/gmapping
  - Hector_slam (Does not need odometry): http://wiki.ros.org/hector_slam
- 3D SLAM
  - RTAB-Map: http://wiki.ros.org/rtabmap_ros, http://introlab.github.io/rtabmap/
  - SLAM with RealSense RGBD camera: https://github.com/IntelRealSense/realsense-ros/wiki/SLAM-with-D435i
- Tf
  - http://wiki.ros.org/tf

# RTAB-Map: Odometry

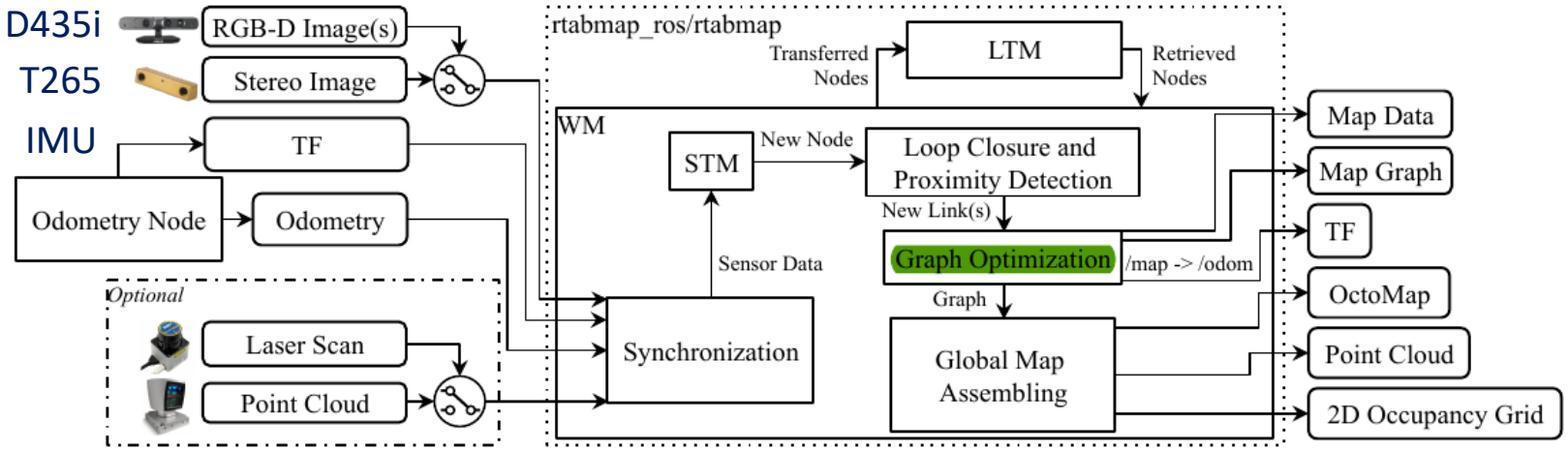
**Visual odometry**

T265 (D435i)

IMU



# RTAB-Map: SLAM




SLAM