Mapping Reconstruction with Livox MID360

This repository provides a comprehensive setup for mapping reconstruction using the Livox MID360 LiDAR sensor. It includes Dockerized environments for easy deployment and tools for data recording, visualization, and processing.

Requirements

Ensure the following requirements are met before proceeding:
- **Ubuntu**: Version 20.04 or higher
- **Docker**: Recommended Video Installation Guide
- **Static IP Address of Host PC**: E.g., `192.168.1.10`
- **Static IP Address of Livox MID360**: E.g., `192.168.1.1`

Results

(Include screenshots, performance metrics, or descriptions of mapping results here.)

Installation & Quick Run of Docker Environment

1. **Clone the Repository**

```
# Create directory for Livox MID360
mkdir livox_mid_360
cd livox_mid_360

# Clone the repository into the directory
git clone git@github.com:djetshu/LivoxMid360-3DScanEnv.git
```

2. **Build the Docker Image**

```
cd LivoxMid360-3DScanEnv

# Build the Docker image
./docker/build.sh
```

> **Note**: The build step is a one-time process. For subsequent runs, proceed directly to the next step.

3. **Run the Docker Container**

```
# Enable permissions for graphics/video
# (Optional: Already in run_docker.sh)
xhost +local:root

# Start the Docker container
./docker/run_docker.sh
```

Quick Running (Inside Docker)

Before running, ensure all configurations are properly set (refer to Set Up and Configuration).

General Instructions

1. Navigate to the workspace:

   ```
   cd /livox_mid_360/livox_mid_360_ws
   ```

2. Compile the ros workspace:

   ```
   colcon build
   ```

3. Source the workspace:

   ```
   source install/setup.bash
   ```

Depending on your use case, you can launch different setups as described below:

Run FastLIO with Livox MID360 ROS Driver

To perform mapping using FastLIO integrated with the Livox MID360:

1. Execute the following command:

   ```
   ros2 launch fast_lio mapping_MID360.launch.py
   ```

Run Only the Livox MID360 ROS Driver

If you only need to run the Livox MID360 ROS driver for data visualization or testing:

1. Launch the required nodes:

   ```
   # To show Data in Rviz2
   ros2 launch livox_ros_driver2 rviz_MID360_launch.py
   # To get data in Livox custom message
   ros2 launch livox_ros_driver2 msg_MID360_launch.py
   ```

Save Data in ROS Bags

To save LiDAR and IMU data in a ROS bag for later analysis or processing:

1. Use the following command, specifying a name for the output bag file:

   ```
   ros2 bag record /livox/imu /livox/lidar -o /livox_mid_360/livox_mid_360_ws/src/rosbag/<
   ```

Set Up and Configuration

Livox MID360 Setup

Modify the `MID360_config.json` file located at `livox_mid_360/src/livox_ros_driver2/config` to set up the IP addresses for the host PC and the Livox MID360.

Example configuration:

```
{
  "lidar_summary_info": {
    "lidar_type": 8
  },
  "MID360": {
    "lidar_net_info": {
      "cmd_data_port": 56100,
      "push_msg_port": 56200,
      "point_data_port": 56300,
      "imu_data_port": 56400,
      "log_data_port": 56500
    },
    "host_net_info": {
      "cmd_data_ip": "192.168.1.5",   # host ip (it can be revised)
      "cmd_data_port": 56101,
      "push_msg_ip": "192.168.1.5",   # host ip (it can be revised)
      "push_msg_port": 56201,
      "point_data_ip": "192.168.1.5", # host ip (it can be revised)
      "point_data_port": 56301,
      "imu_data_ip": "192.168.1.5",   # host ip (it can be revised)
      "imu_data_port": 56401,
      "log_data_ip": "",
      "log_data_port": 56501
    }
  },
  "lidar_configs": [
    {
      "ip": "192.168.1.12",   # ip of the LiDAR you want to config
      "pcl_data_type": 1,
      "pattern_mode": 0,
      "extrinsic_parameter": {
        "roll": 0.0,
        "pitch": 0.0,
        "yaw": 0.0,
        "x": 0,
        "y": 0,
        "z": 0
      }
```

```
    }
  ]
}
```

Notes:

- Update the placeholder IP addresses (`192.168.1.5` and `192.168.1.12`) with the actual static IP addresses of your host PC and Livox MID360.

References

This repository contains the following packages: - Livox SDK2 - Livox Ros Driver - Fast LIO