

Declare and read Canny thresholds

Description : Add `declare_parameter` entries for `lower_threshold` (default 100) and `upper_threshold` (default 200) in the `data_parameters` camera node, then call `can` build and run the node.

Use `get_parameter_as<double>` in the node to read and log both values at startup to confirm they load correctly.

Open RQT and locate dynamic parameters

Description : Run `ros2 run rqt_gui rqt`, navigate to Plugins → Configuration → Dynamic Reconfigure, and find the camera node's parameters.

Change the lower/upper threshold values there and verify the node accepts the updates without restarting.

Live-tune Canny while viewing the camera feed

Description : With the camera node running against the kitty data set, adjust the Canny `lower_threshold` and `upper_threshold` in RQT and watch the live image output.

Observe immediate visual changes (more or fewer edges) and note parameter values that produce the desired result.

Add and test a cropping parameter X

Description : Declare a parameter named `X` in the camera node to control a crop region (e.g., x offset or width), then call `can` build and run the node to enable cropping.

Modify `X` via RQT and confirm the camera feed updates in real time to show the area-of-interest change.

Tune voxel-grid and ROI for LIDAR point cloud

Description : Expose voxel grid parameters (`leaf_size`) and X/Y/Z min/max bounds in the LIDAR processing node, build and run the pipeline subscribing to the processed point cloud.

Use RQT to change `leaf_size` and min/max limits while viewing the processed cloud in Rviz to see density and ROI segmentation update live.