laser pipeline

⇒ Meta-package of libraries for processing laser data, including converting laser data into 3D representations.

- laser_assembler
- laser_filters
- laser_geometry

★ laser filter

⇒ Assorted filters designed to operate on 2D planar laser scanners, which use the sensor_msgs/LaserScan type.

⇒ The primary content of the laser_filters package is a number of general purpose filters for processing sensor_msgs/LaserScan messages.

⇒ This package provides two nodes that can run multiple filters internally.

- → The scan_to_scan_filter_chain applies a series of filters to a sensor_msgs/LaserScan.
- → The scan_to_cloud_filter_chain first applies a series of filters to a sensor_msgs/LaserScan, transforms it into a sensor_msgs/PointCloud, and then applies a series of filters to the sensor_msgs/PointCloud.

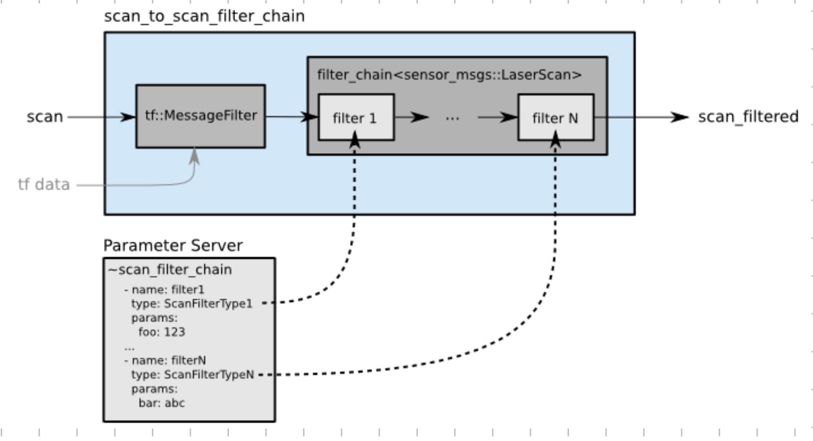
⇒ Each laser filter is a separate plugin exported by the laser_filters package.

⇒ This allows them to be specified in a configuration file which can be loaded into an arbitrary filter_chain

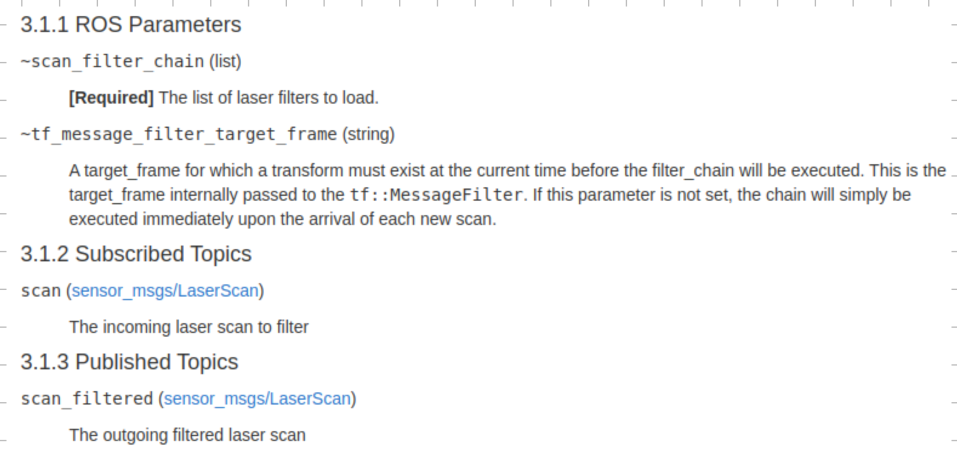
- → Filter chains are configured from the parameter server.
- → Each filter specified in the chain will be applied in order.

⇒ The individual filters configurations contain:

- → a name which is used for debugging purposes,
- → a type which is used to locate the plugin,
- → and a params which is a dictionary of additional variables.



```
~scan_filter_chain
  - name: filter1
    type: ScanFilterType1
    params:
      foo: 123
  ...
  - name: filterN
    type: ScanFilterTypeN
    params:
      bar: abc
```



3.1.1 ROS Parameters

~scan_filter_chain (list)

[Required] The list of laser filters to load.

~tf_message_filter_target_frame (string)

A target_frame for which a transform must exist at the current time before the filter_chain will be executed. This is the target_frame internally passed to the `tf::MessageFilter`. If this parameter is not set, the chain will simply be executed immediately upon the arrival of each new scan.

3.1.2 Subscribed Topics

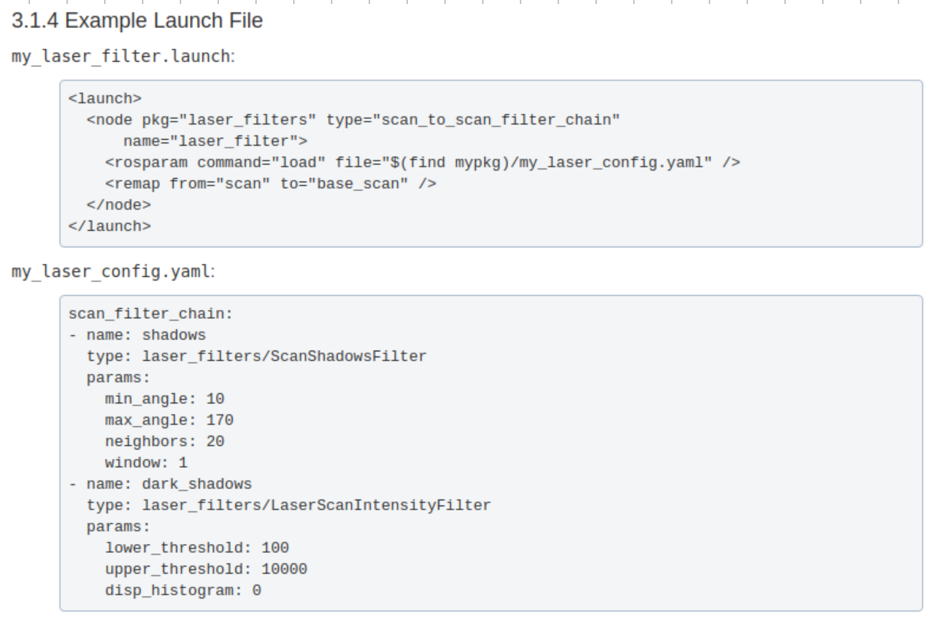
`scan` (sensor_msgs/LaserScan)

The incoming laser scan to filter

3.1.3 Published Topics

`scan_filtered` (sensor_msgs/LaserScan)

The outgoing filtered laser scan



3.1.4 Example Launch File

`my_laser_filter.launch`:

```
<launch>
  <node pkg="laser_filters" type="scan_to_scan_filter_chain"
      name="laser_filter">
    <rosparam command="load" file="$(find mypkg)/my_laser_config.yaml" />
    <remap from="scan" to="base_scan" />
  </node>
</launch>
```

`my_laser_config.yaml`:

```
scan_filter_chain:
- name: shadows
  type: laser_filters/ScanShadowsFilter
  params:
    min_angle: 10
    max_angle: 170
    neighbors: 20
    window: 1
- name: dark_shadows
  type: laser_filters/LaserScanIntensityFilter
  params:
    lower_threshold: 100
    upper_threshold: 10000
    disp_histogram: 0
```

★ laser_filter plugins

① LaserArrayFilter