Basic perception

Olivier Aycard

Professor

Grenoble INP - PHELMA

GIPSA Lab

https://www.gipsa-lab.grenoble-inp.fr/user/olivier.aycard

olivier.aycard@grenoble-inp.fr

Lab on basic perception

- To be sure, you have the last release of the code, download it on git:

1. Cd ~/catkin_ws/src
2. \rm –r follow_me
3. Git clone https://gricad-gitlab.univ-grenoble-alpes.fr/aycardol/follow_me.git

Lab on basic perception

- We will implement and test:
  1. The detection of motion algorithm described during the lecture on DATMO;
  2. The clustering of objects algorithm described during the lecture on DATMO.

Detection of motion

1. Edit and modify *datmo.cpp* in *~/catkin_ws/src/follow_me*;
   1. You should implement and understantd the functions « store_background », « reset_motion » and « detect_current_motion »
   2. You should implement the function « detect_motion » and complete what robair should do in the 4 different cases ;
2. Edit datmo.h to see the data structure and prototypes of functions
3. Edit *detection_node.cpp* in *~/catkin_ws/src/follow_me*;
   - ➢ You should have a look on the source file;
   - ➢ This is the main node that will be used to detect a moving person ;
4. Check the results in a terminal and rviz

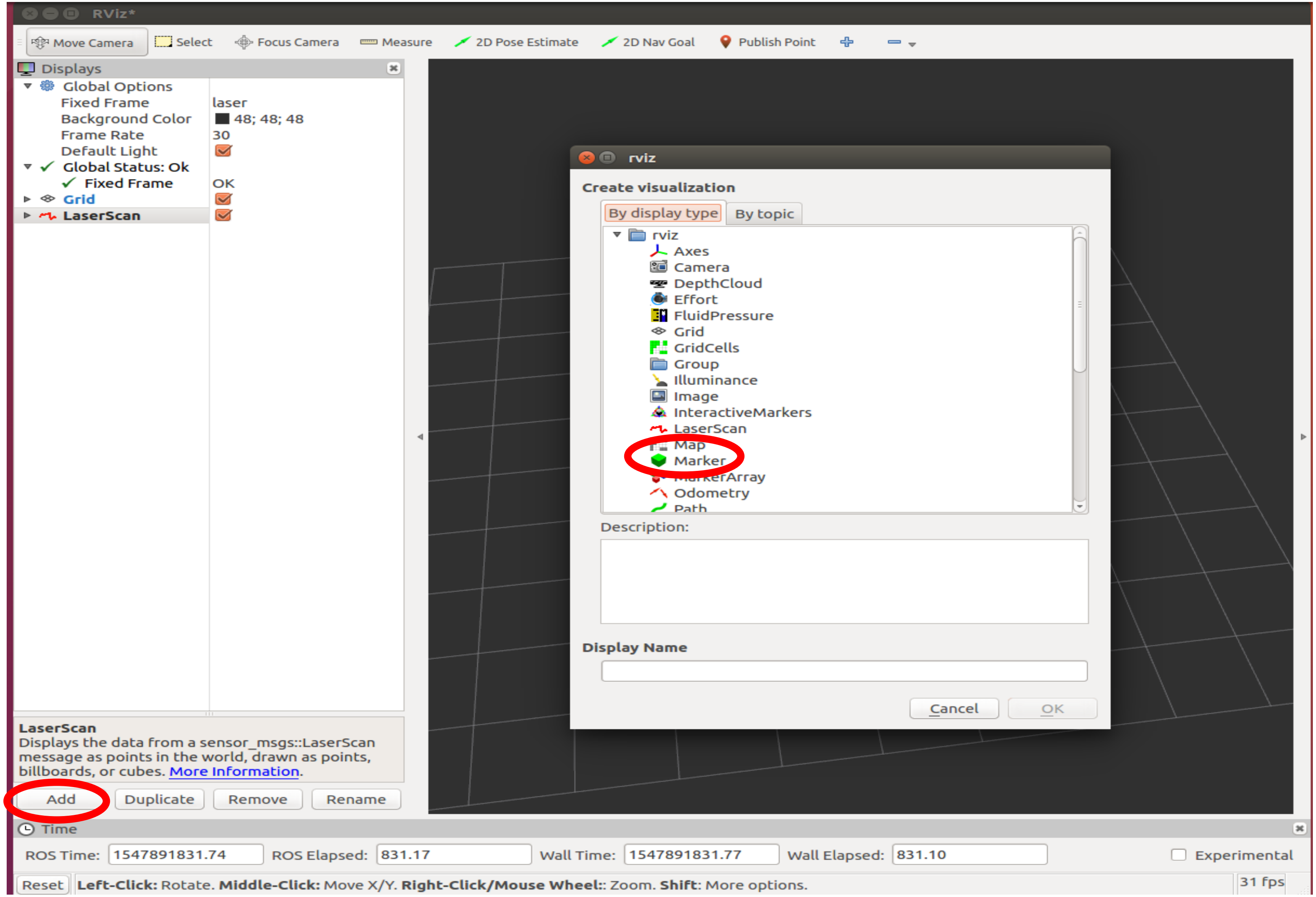
Tests(1/3)

- Open 5 terminals:

1. Roscore: the ROS master;
2. Rosbag play *data_file*.bag: to play a saved file;
3. Rosrun follow_me *detection_node*;
4. Rosrun follow_me *robot_moving_node*;
   - The laser data are only processed when the robot does not move;
   - This is automatically taken into account by the node *robot_moving_node*;
   - You do not have to take care about this issue.

5. Rviz: the vizualization tool of ROS.
   - To have a graphical display of the processing;
   - See screenshots on next slides

Tests(2/3)



Global Options
Fixed Frame: laser
Background Color: ■ 48; 48; 48
Frame Rate: 30
Default Light: ☑
✓ Global Status: Ok
✓ Fixed Frame: OK
Grid: ☑
LaserScan: ☑

LaserScan
Displays the data from a sensor_msgs::LaserScan message as points in the world, drawn as points, billboards, or cubes. More Information.

rviz
- Axes
- Camera
- DepthCloud
- Effort
- FluidPressure
- Grid
- GridCells
- Group
- Illuminance
- Image
- InteractiveMarkers
- LaserScan
- Map
- Marker
- MarkerArray
- Odometry
- Path

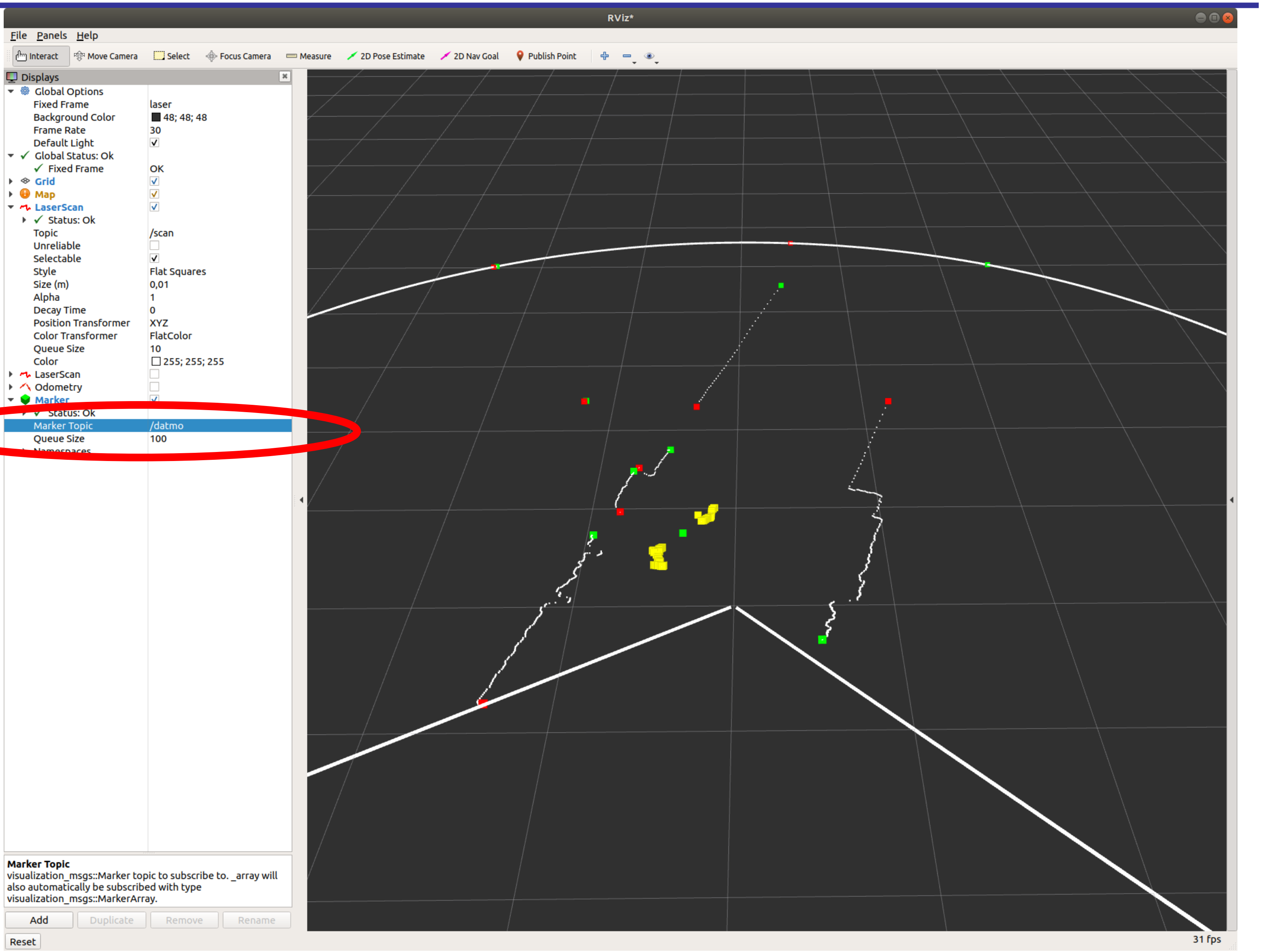
Description:

Display Name

ROS Time:	1547891831.74
ROS Elapsed:	831.17
Wall Time:	1547891831.77
Wall Elapsed:	831.10

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options.

Tests(3/3)



```
Global Options
    Fixed Frame            laser
    Background Color       48; 48; 48
    Frame Rate             30
    Default Light          ☑
Global Status: Ok
    Fixed Frame            OK
Grid                       ☑
Map                        ☑
LaserScan                  ☑
    Status: Ok
    Topic                  /scan
    Unreliable             □
    Selectable             ☑
    Style                  Flat Squares
    Size (m)               0,01
    Alpha                  1
    Decay Time             0
    Position Transformer   XYZ
    Color Transformer      FlatColor
    Queue Size             10
    Color                  255; 255; 255
LaserScan                  □
Odometry                   □
Marker                     ☑
    Status: Ok
    Marker Topic           /datmo
    Queue Size             100
    Namespaces
```

Marker Topic
visualization_msgs::Marker topic to subscribe to. _array will also automatically be subscribed with type visualization_msgs::MarkerArray.

aycard@imag.fr

Lab on basic perception

- We will implement and test:

1. The detection of motion algorithm described during the lecture on DATMO;
2. The clustering of objects algorithm described during the lecture on DATMO.

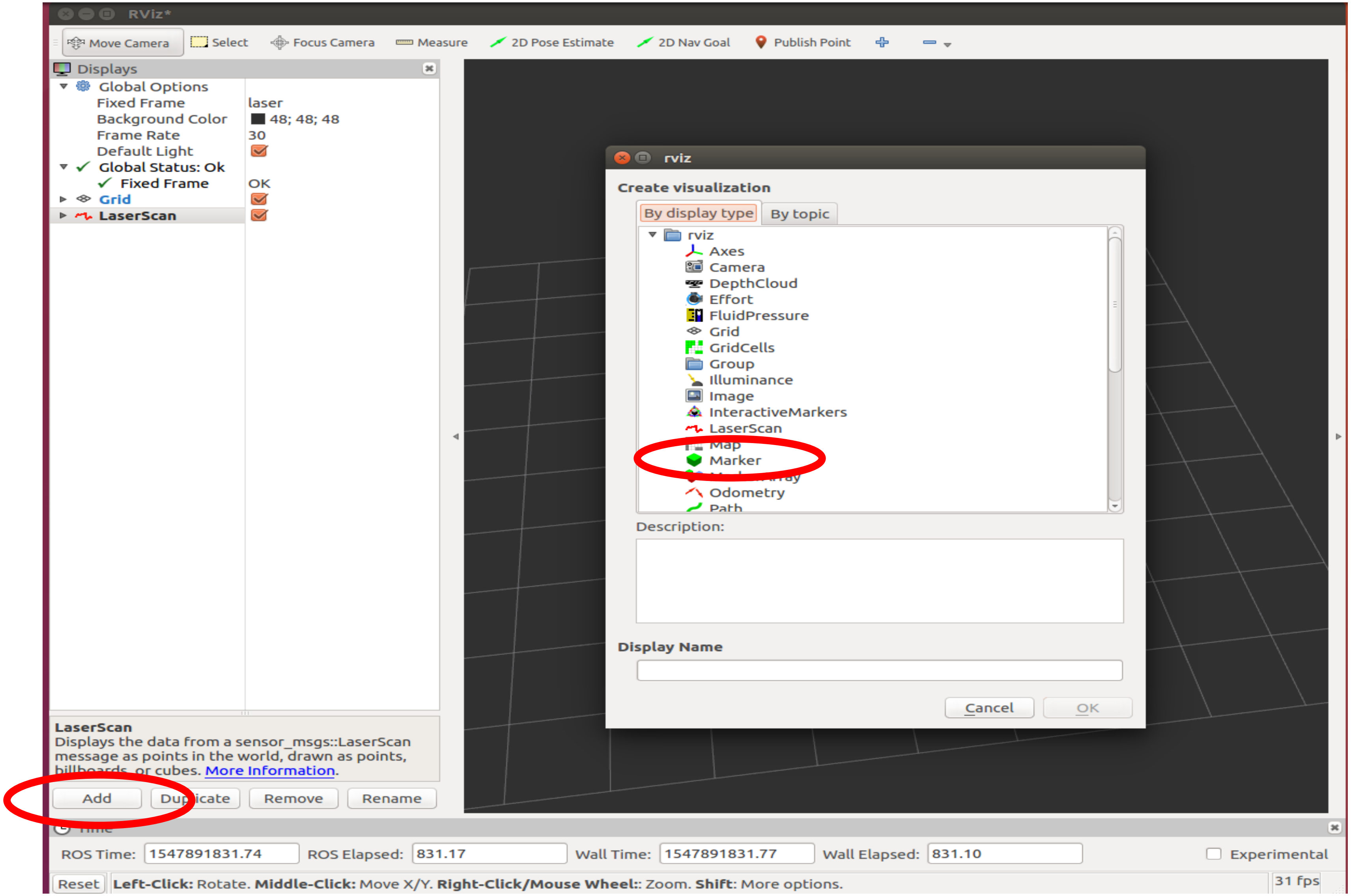
Clustering

1. Edit and modify *datmo.cpp* in *~/catkin_ws/src/follow_me*;
    1. You should implement the function « perform_basic_clustering »
2. Edit datmo.h to see the data structure and prototypes of functions
3. Check the results in a terminal and rviz

Tests(1/3)

- Open 4 terminals:
  1. Roscore: the ROS master;
  2. Rosbag play *data_file*.bag: to play a saved file;
  3. Rosrun follow_me *perform_clustering_node*;
  4. Rviz: the vizualization tool of ROS.
     - ➢ To have a graphical display of the processing;
     - ➢ See screenshots on next slides

Tests(2/3)



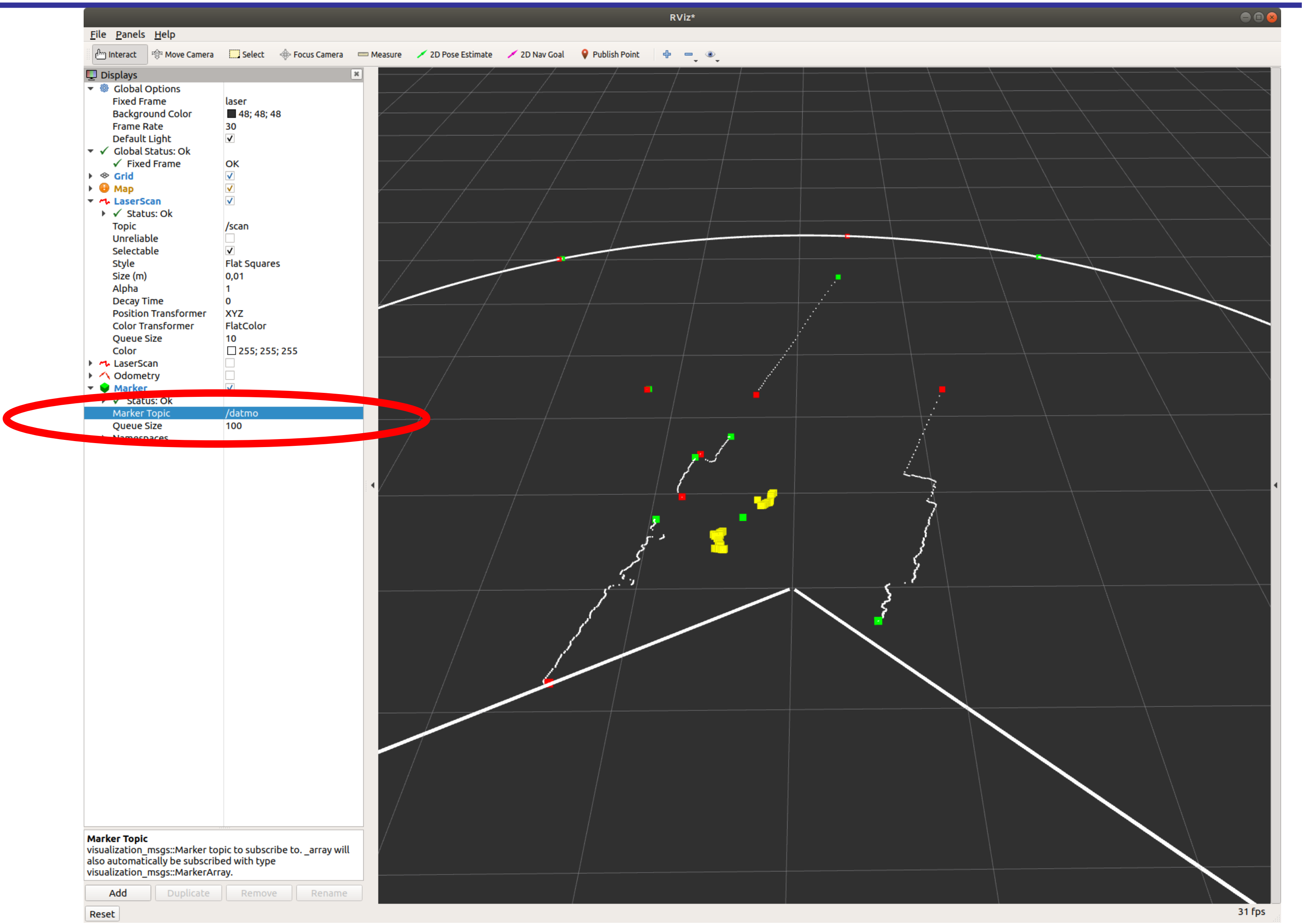
Global Options
Fixed Frame: laser
Background Color: ■ 48; 48; 48
Frame Rate: 30
Default Light: ☑
✓ Global Status: Ok
✓ Fixed Frame: OK
Grid ☑
LaserScan ☑

LaserScan
Displays the data from a sensor_msgs::LaserScan message as points in the world, drawn as points, billboards, or cubes. More Information.

ROS Time: 1547891831.74 ROS Elapsed: 831.17 Wall Time: 1547891831.77 Wall Elapsed: 831.10 □ Experimental

Reset **Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options. 31 fps

Tests(3/3)



Global Options
Fixed Frame laser
Background Color ■ 48; 48; 48
Frame Rate 30
Default Light ☑
✓ Global Status: Ok
✓ Fixed Frame OK
Grid ☑
Map ☑
LaserScan ☑
✓ Status: Ok
Topic /scan
Unreliable ☐
Selectable ☑
Style Flat Squares
Size (m) 0,01
Alpha 1
Decay Time 0
Position Transformer XYZ
Color Transformer FlatColor
Queue Size 10
Color □ 255; 255; 255
LaserScan ☐
Odometry ☐
Marker ☑
✓ Status: Ok
Marker Topic /datmo
Queue Size 100
Namespaces

Marker Topic
visualization_msgs::Marker topic to subscribe to. _array will also automatically be subscribed with type visualization_msgs::MarkerArray.