Displaying Atlas' LIDAR Data in Rviz

Wyatt Newman
October, 2014

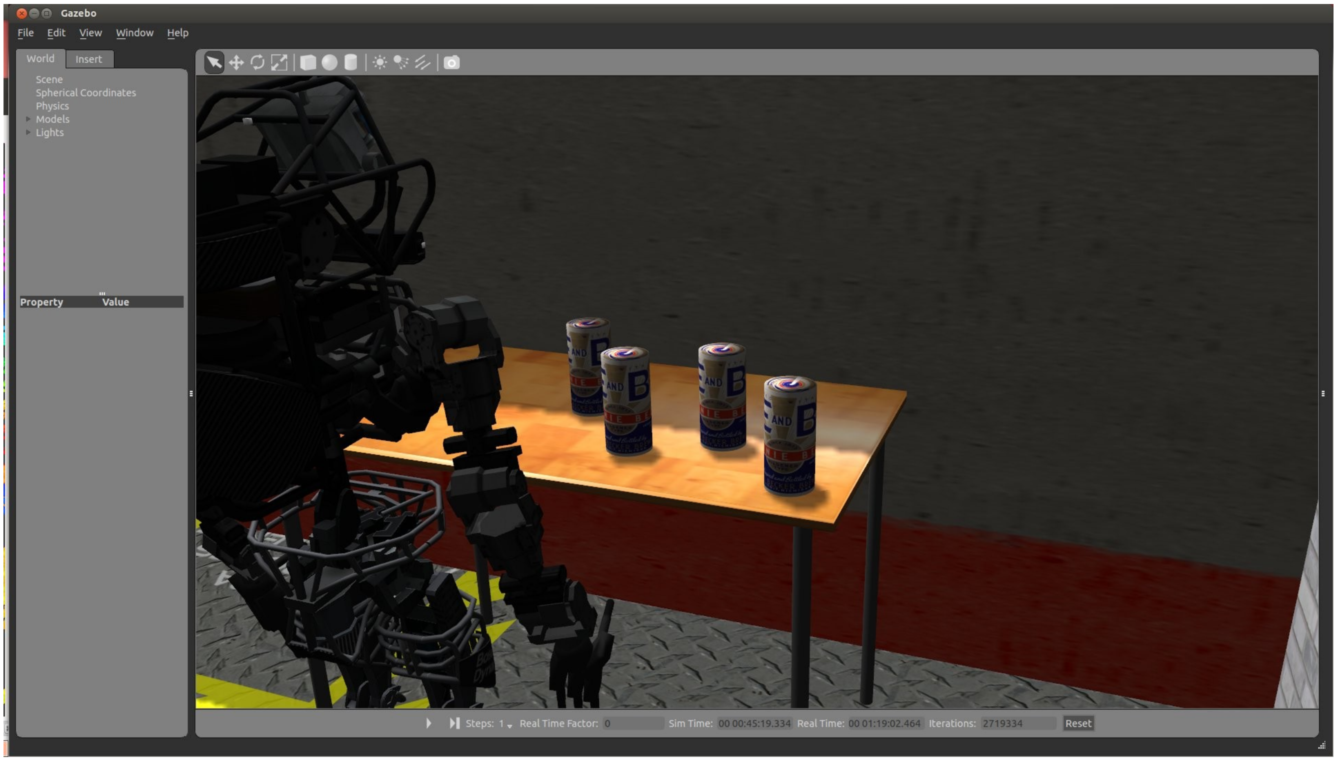
We have seen two ways to view simulated Atlas: via Gazebo and via Rviz. It is important to understand the distinction between these. Gazebo/drcsim is a simulator, meant to act as a substitute for a real robot. When controlling the actual, physical Atlas robot in the lab, the Gazebo display will not exist. The same topics will be available, such as atlas/atlas_state, but these will be published by the actual robot, not the robot simulator.

In development of software for Atlas, the Gazebo view can be useful for interpreting and debugging results. However, one should avoid depending on this view for developing operator interfaces, since this view will not exist for the real robot. Instead, one can use a display of the sensor values published by Atlas. This is conveniently achieved using the Rviz visualization tool.

The screen capture below is from a modification of DRC task 6 (picking up a drill). In Gazebo, the "world" file was modified to remove the drill and bring in several cans, as well as to reposition Atlas conveniently in front of the table. This world can be launched with the command:

```
roslaunch hku_worlds beer_table.launch
```

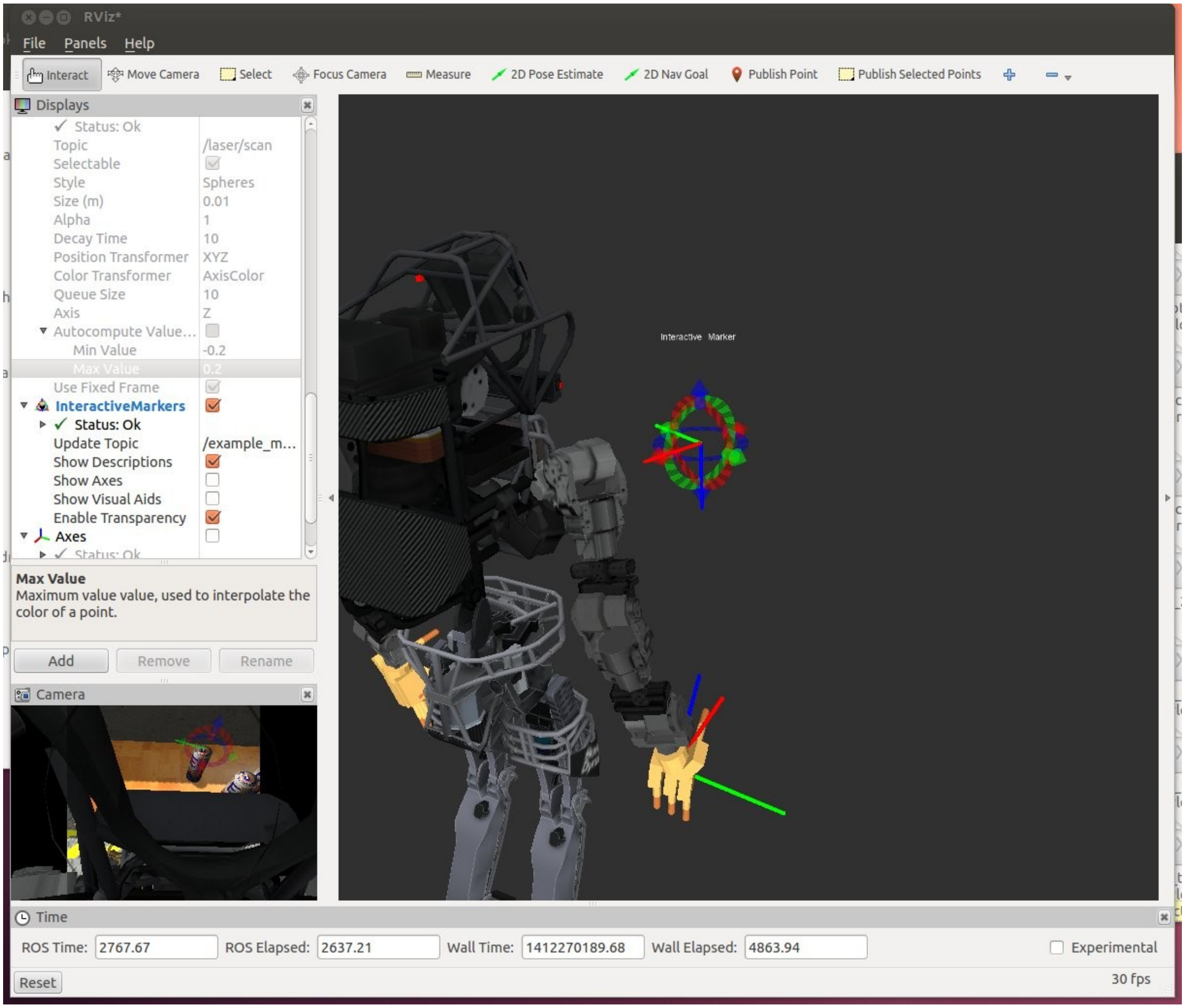
This launch file uses the "world" file in hku_worlds/worlds/beer_table.world



Steps: 1 Real Time Factor: 0 Sim Time: 00 00:45:19.334 Real Time: 00 01:19:02.464 Iterations: 2719334

This view would not be available to a remote operator of physical Atlas, be we can instead bring up "Rviz": rosrun rviz rviz.

The screenshot below shows a model of Atlas that is posed to match the joint data from atlas/atlas_state. As a result, the rviz view looks similar to the Gazebo view. In addition, this view has turned on display of an interactive marker (from the example in package: example_interactive_marker). It also shows the right_grasp_frame axes, as well as the emulated camera view from Atlas's sensor head, left camera. This can be seen in the small, lower-left panel of the rviz screen capture. Note that the interactive marker also shows up in this view, which can be convenient for an operator interface. However, the main display does not show the table with cans. This is because the main display is a 3-D rendering, whereas the camera display is merely a 2-D image. They are thus not compatible, until/unless the camera data is somehow augmented with depth information.



✓ Status: Ok
Topic /laser/scan
Selectable ☑
Style Spheres
Size (m) 0.01
Alpha 1
Decay Time 10
Position Transformer XYZ
Color Transformer AxisColor
Queue Size 10
Axis Z
Autocompute Value... □
Min Value -0.2
Max Value 0.2
Use Fixed Frame ☑
InteractiveMarkers ☑
✓ Status: Ok
Update Topic /example_m...
Show Descriptions ☑
Show Axes □
Show Visual Aids □
Enable Transparency ☑
Axes □
✓ Status: Ok

Max Value
Maximum value value, used to interpolate the color of a point.

ROS Time: 2767.67 ROS Elapsed: 2637.21 Wall Time: 1412270189.68 Wall Elapsed: 4863.94 □ Experimental

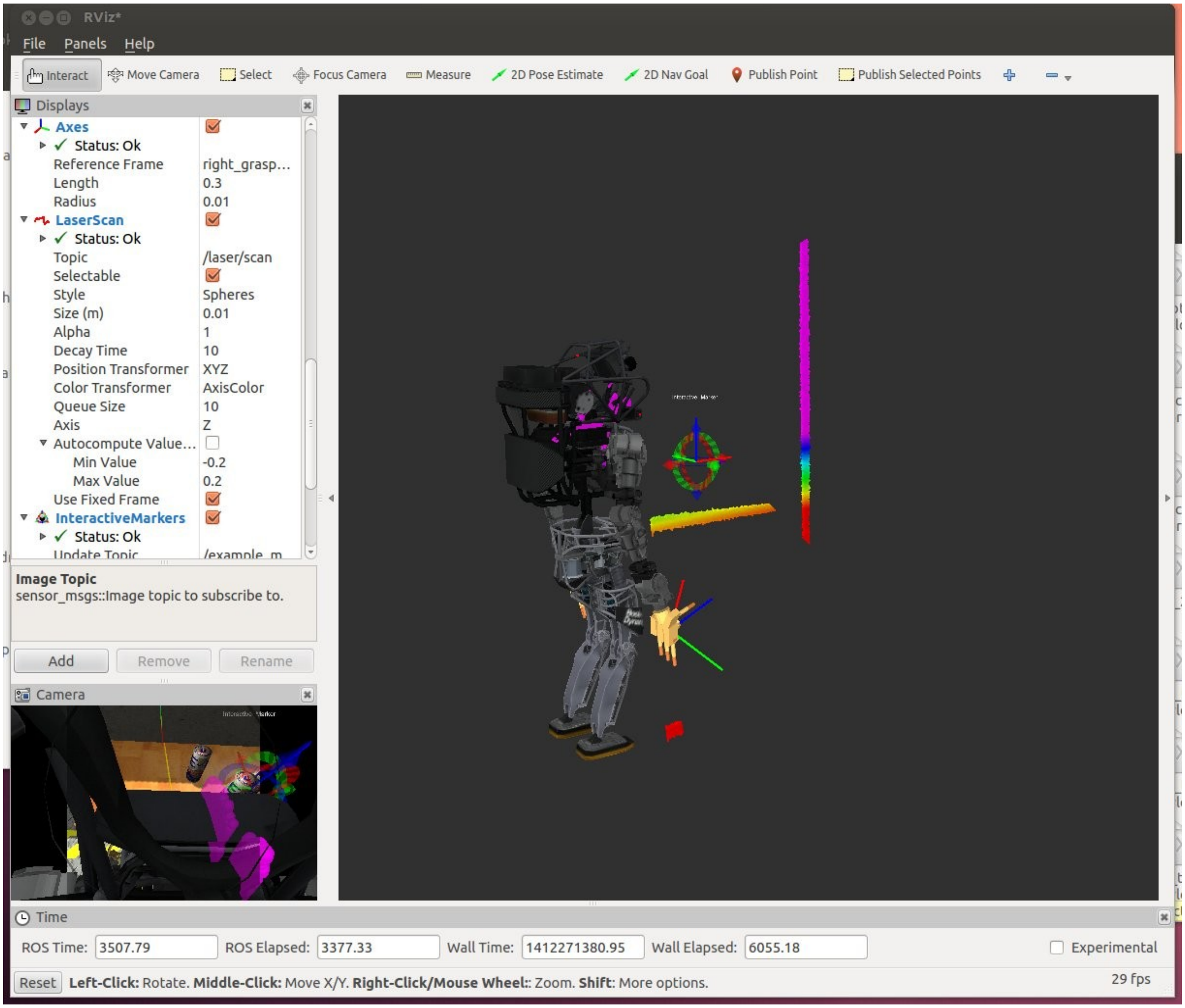
30 fps

The sensor-head's LIDAR, however, *does* offer 3-D sensory data, and we can display this data in Rviz.

The image below includes display of Atlas's LIDAR data. This is accomplished by the following steps, performed interactively with Rviz via the "Displays" panel.

- In Rviz, click "Add", then choose "LaserScan". A new item will appear in the "Displays" panel, called "LaserScan"
- Expand this item to expose its properties.
- Set the "Topic" to /laser/scan (This is the ROS topic to which the LIDAR publishes its data)
- Optionally, set "Style" to: Spheres
- Optionally, set Color Transformer to: AxisColor, and
- Set Axis to: Z
- "uncheck" "Autocompute Value" and you will be able to adjust the min and max range over which the LIDAR points will be colorized as a function of distance.

These options can be set (and adjusted) to achieve a more intuitive use of color for displaying the LIDAR points. The resulting display will be something like that below.



```
Axes                    ☑
  ✓ Status: Ok
  Reference Frame       right_grasp...
  Length                0.3
  Radius                0.01
LaserScan               ☑
  ✓ Status: Ok
  Topic                 /laser/scan
  Selectable            ☑
  Style                 Spheres
  Size (m)              0.01
  Alpha                 1
  Decay Time            10
  Position Transformer  XYZ
  Color Transformer     AxisColor
  Queue Size            10
  Axis                  Z
  Autocompute Value...  □
    Min Value           -0.2
    Max Value           0.2
  Use Fixed Frame       ☑
InteractiveMarkers      ☑
  ✓ Status: Ok
  Update Topic          /example_m
```

Image Topic
sensor_msgs::Image topic to subscribe to.

Time

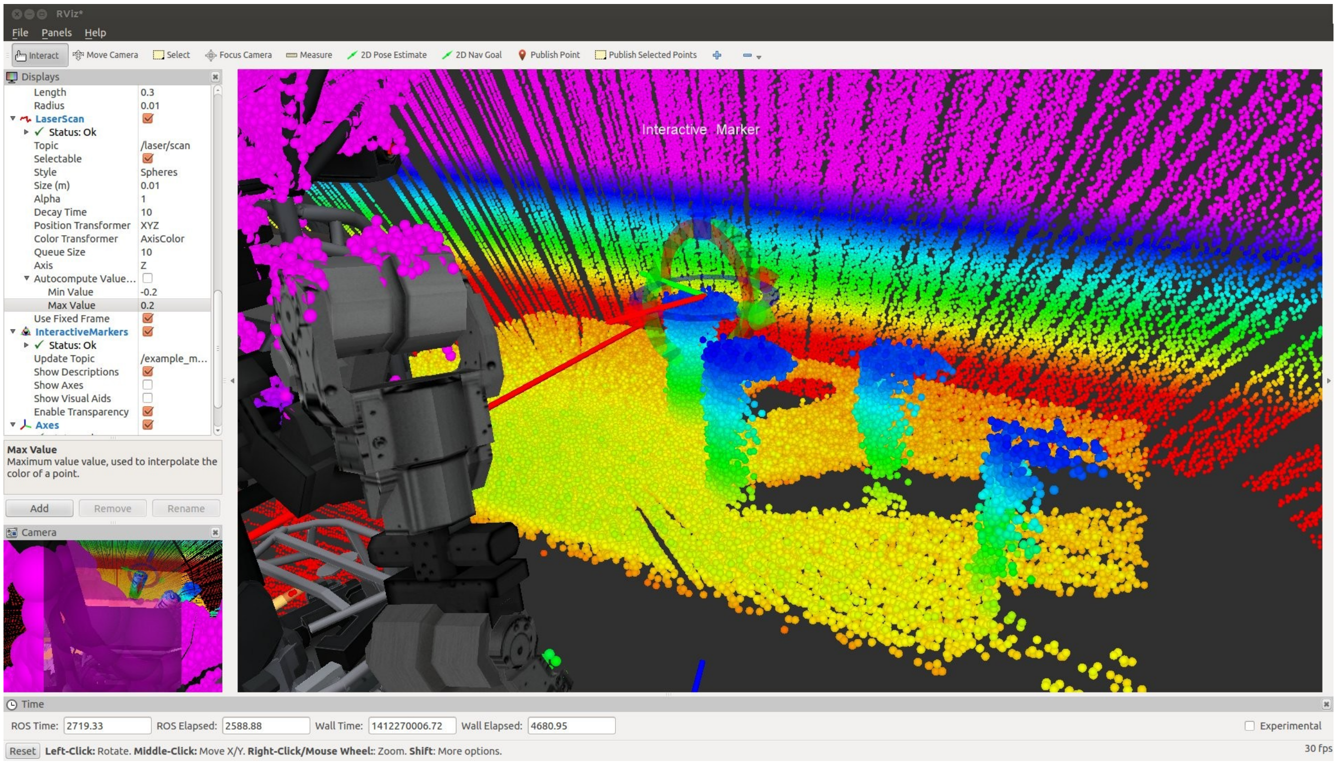
ROS Time: 3507.79 ROS Elapsed: 3377.33 Wall Time: 1412271380.95 Wall Elapsed: 6055.18 □ Experimental

Reset **Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options. 29 fps

In the above display, the 3-D LIDAR points only appear along one “stripe” in space. This is because Atlas's LIDAR is not yet rotating. To induce rotation of the LIDAR, enter the following command:

```
rostopic pub --once /multisense_sl/set_spindle_speed std_msgs/Float64 '{data: 3.0}'
```

The above command causes the LIDAR to spin at 3.0 radians/sec. In Rviz, this will result in a dynamic display of LIDAR swept points. However, it is hard to interpret the environment from these points, since they appear only transiently. To build up a display that includes numerous LIDAR scan slices, set the LaserScan parameter “Decay Time” to a non-zero number, e.g. “10” (seconds). The resulting display will be something like that below.



Length	0.3
Radius	0.01
▾ LaserScan	☑
▸ ✓ Status: Ok	
Topic	/laser/scan
Selectable	☑
Style	Spheres
Size (m)	0.01
Alpha	1
Decay Time	10
Position Transformer	XYZ
Color Transformer	AxisColor
Queue Size	10
Axis	Z
▾ Autocompute Value...	□
Min Value	-0.2
Max Value	0.2
Use Fixed Frame	☑
▾ InteractiveMarkers	☑
▸ ✓ Status: Ok	
Update Topic	/example_m...
Show Descriptions	☑
Show Axes	□
Show Visual Aids	□
Enable Transparency	☑
▾ Axes	☑

Max Value
Maximum value value, used to interpolate the color of a point.

ROS Time: 2719.33 ROS Elapsed: 2588.88 Wall Time: 1412270006.72 Wall Elapsed: 4680.95 □ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options. 30 fps

In the above scene, the colorization was adjusted to visualize elevation (Z height), and the range was adjusted to highlight elevations from the table top to the top of the cans. From this view, it is easy to see the can tops (in blue). It can also be seen that the interactive marker is aligned with its origin at the top of one of the cans, with the z-axis (e.g., desired palm normal direction) pointing down, through the can. This would be a potential pose for grasping the can from above.

The operator can rotate views in rviz and move the interactive marker until a desirable alignment is achieved.

From the above scene, one can get a sense of the sampling resolution and sensor noise of the LIDAR sensor. Interpretation of the display can be non-intuitive, since the points displayed are only the points available from the sensor head. Thus, the backs of the cans are not visible. Further, there appear to be "holes" in the table top, as these missing points are shadowed by the cans.

One further rviz tool is valuable for interacting with 3-D data. The rviz "plug-in" called "Publish Selected Points" can be added to the Rviz toolbar. To do so, perform these steps:

1. Press "+" on the rviz toolbar. A tool menu will appear.
2. Select "rviz_plugin_selected_points_topic->PublishSelectedPoints". The tool will then appear on the toolbar as "Publish Selected Points"

If you save your rviz configuration when you close rviz, you will not need to repeat these steps in the future; the PublishSelectedPoints tool will remain available in the toolbar.

To use this tool, click on the new tool "publish_selected_points" in the rviz menu bar . (It will become highlighted to show it has been selected). Then, left-click/drag to select points in rviz. The region of points selected will get published to the topic: /selected_points. You can see that this is working by entering the command:

```
rostopic echo /selected_points
```

This data is thus available to nodes that you can develop to interpret the 3-D data.