

[← Back \(/tutorials?cat=\)](/tutorials?cat=)

# Intermediate: Model Appearance

---

## Improve Appearance

Models with textures and 3D meshes can improve your visual experience, and more importantly improve the realism of an environment. Simulated cameras that feed information to vision processing algorithms will benefit from models that appear realistic as well.

In this section, we will use 3D meshes available on the Velodyne website to improve the visual appearance of our model. More manufactures are making 3D meshes available, however it can sometimes be difficult to find an existing mesh. In these cases you can try your hand at mesh creation, work with an artist, or contact the manufacturer directly.

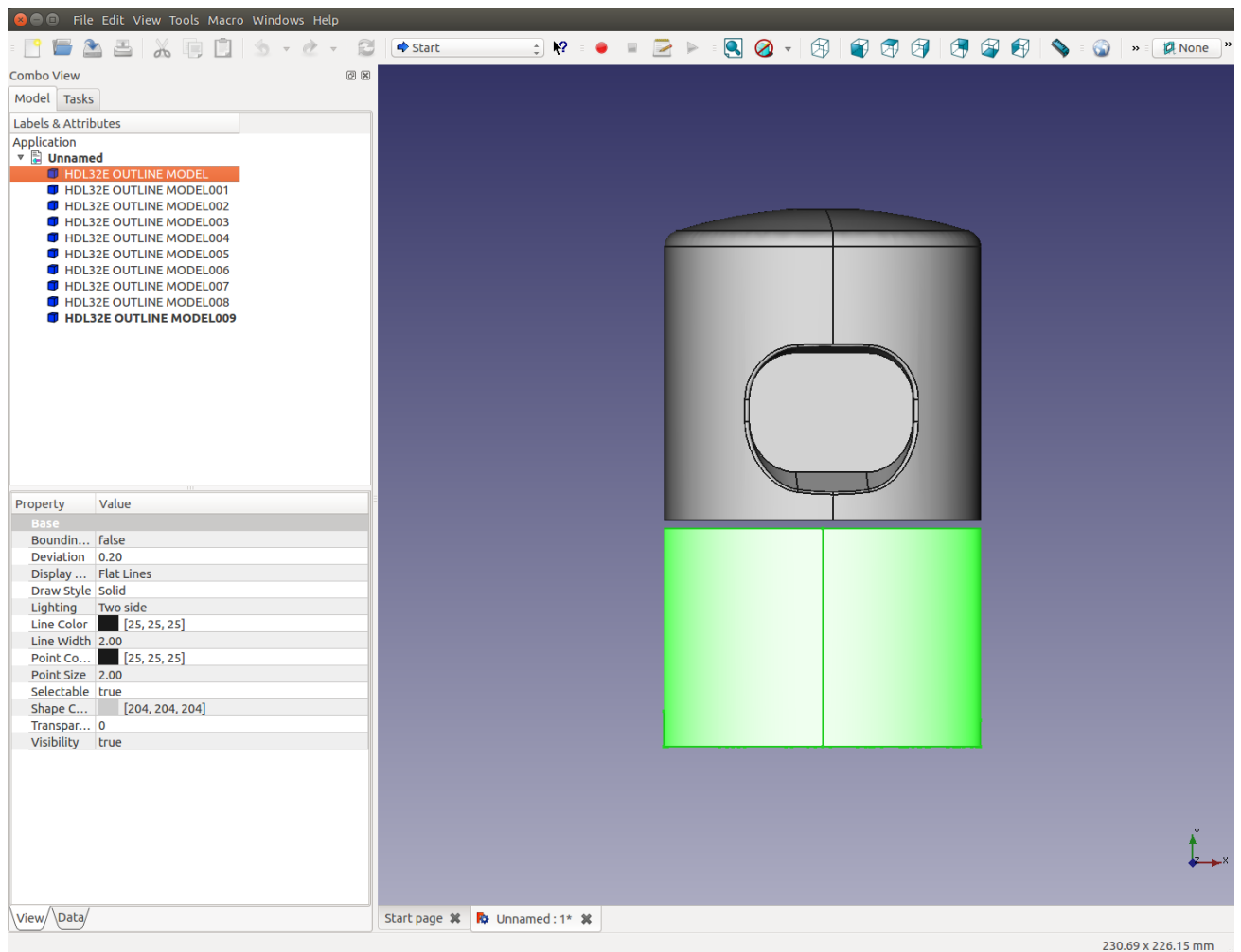
Velodyne has a STEP file ([http://velodynelidar.com/docs/drawings/HDL32E\\_Outline\\_Model.STEP](http://velodynelidar.com/docs/drawings/HDL32E_Outline_Model.STEP)) for the HDL-32 located on their website (<http://velodynelidar.com>). Gazebo can only use STL, OBJ or Collada files, so we'll have to convert this file and then add it to our model.

## Step 1: Mesh Acquisition

1. Download the STEP file. Right-click and save-as on this link ([http://velodynelidar.com/docs/drawings/HDL32E\\_Outline\\_Model.STEP](http://velodynelidar.com/docs/drawings/HDL32E_Outline_Model.STEP)) to save the STEP file.
2. Open the STEP file in FreeCad. If you're using Ubuntu, you can install freecad using: `sudo apt-get install freecad`

```
freecad ~/Downloads/HDL32E_Outline_Model.STEP
```

3. Select the base of Velodyne, by clicking on "HDL32 OUTLINE MODEL" in the left-hand **Labels & Attributes** panel.



4. Export to Collada, into a file called `velodyne_base.dae` .

**File->Export**

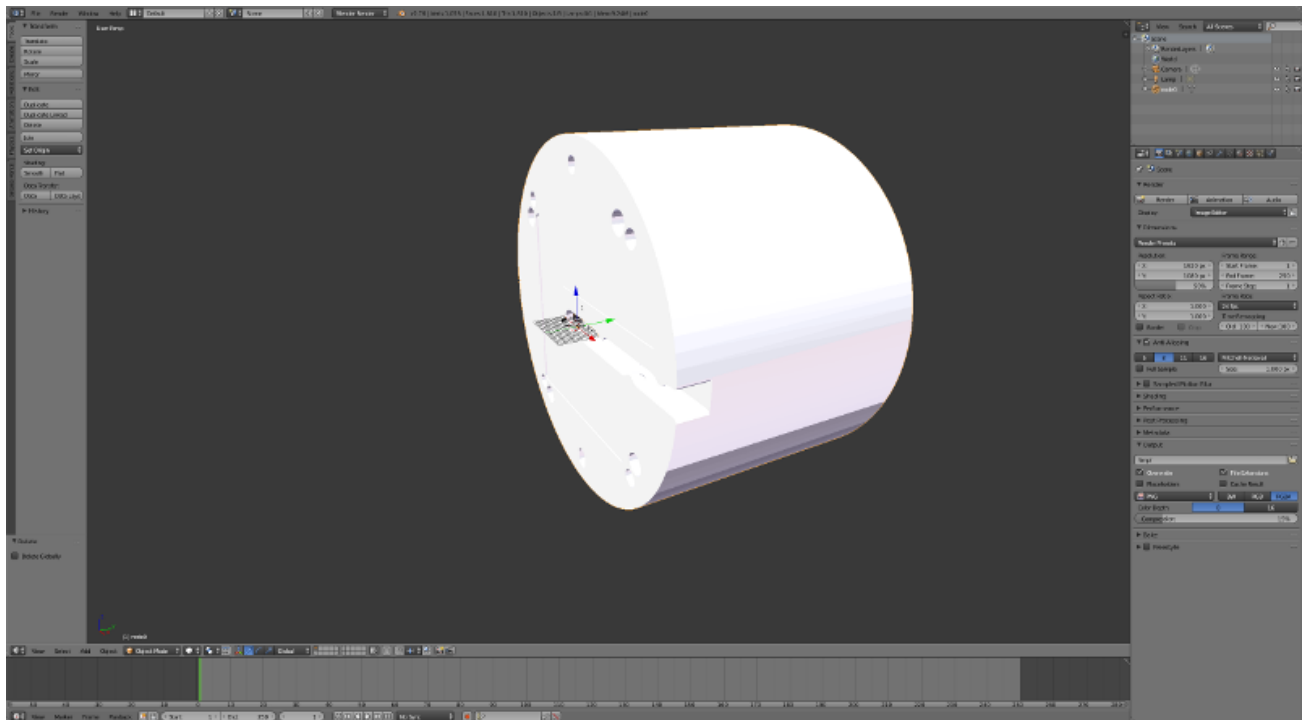
5. We need to modify the `velodyne_base.dae` file in Blender (<https://www.blender.org/>), because the units are incorrect and we want the mesh centered on the origin.

blender

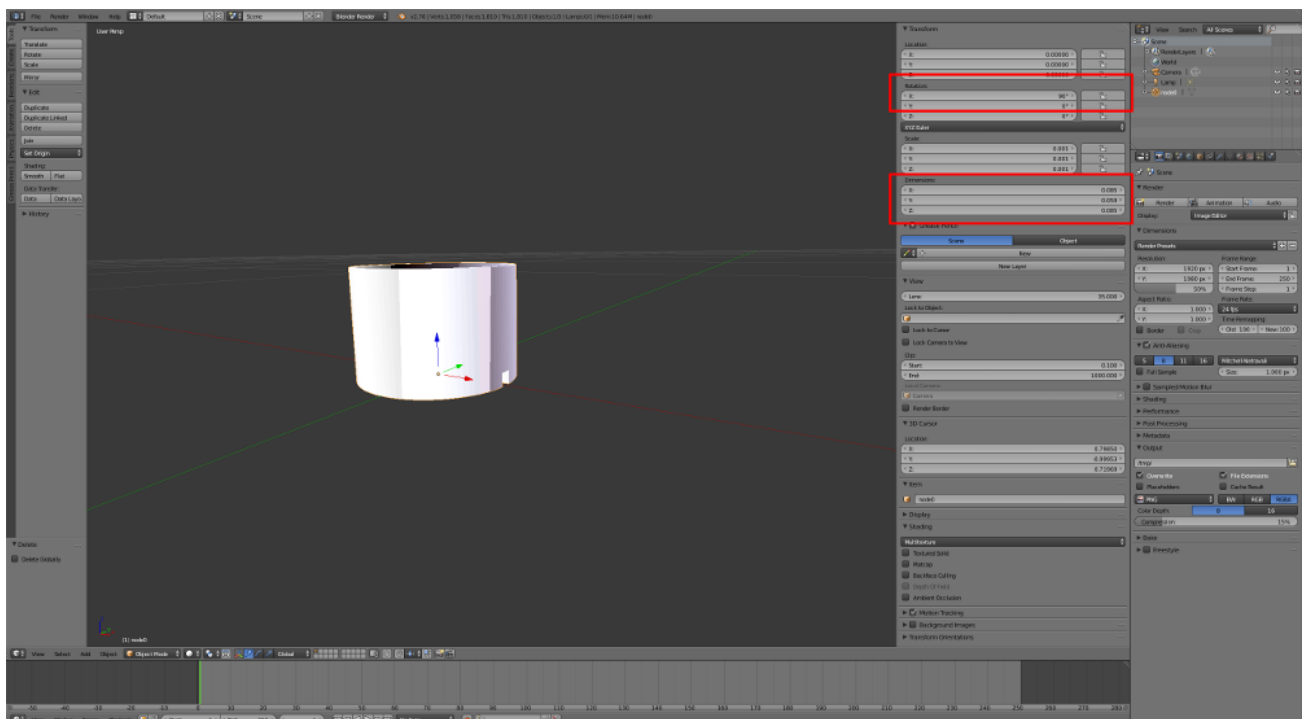
6. Import the `velodyne_base.dae` file.

**File->Import->Collada**

Note: You may have to download a newer version of Blender (<https://www.blender.org/>) to get the Collada import feature.



7. The units are currently in millimeters, and Gazebo requires meters. The model is also rotated so that the top is facing along the Y-axis, and we would like the top to face along the Z-axis.
8. Pull out the right-tab in blender (look for a plus sign near the upper right of the render window). Under the **Dimensions** section of this tab, divide the x,y,z components by 1000. See image below.
9. In the same tab, rotate the model by 90 degrees around the X-axis. See image below.
10. The mesh should not look like the following image.

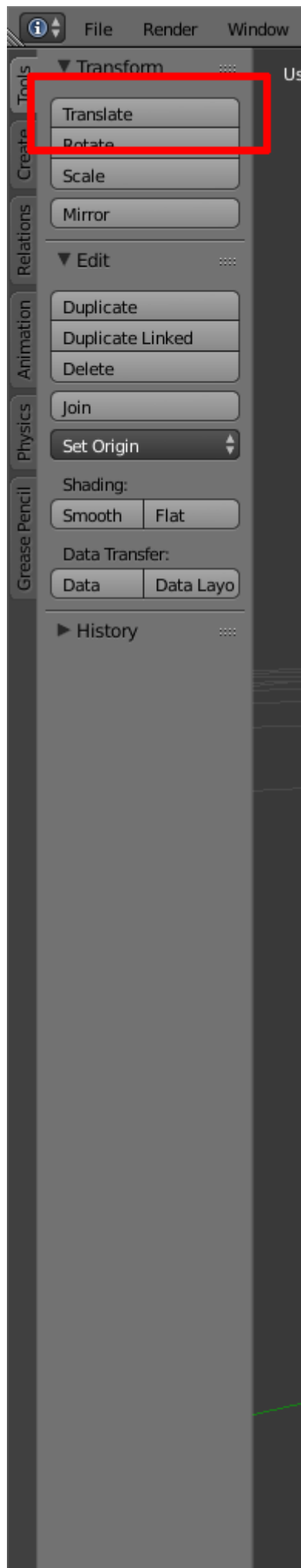


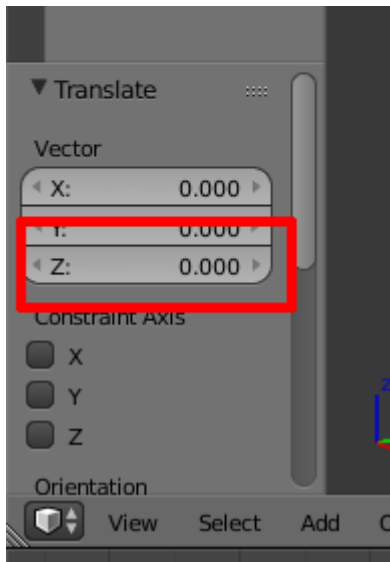
11. Export the mesh as a collada file.

**File->Export->Collada**

12. Repeat this process for the top of the velodyne. On FreeCAD, export "HDL32E OUTLINE MODEL006" as `velodyne_top.dae` . You will also have to translate this mesh so that the bottom is on the XY-plane. Use the **Translate** button in the upper left (click on it twice to open a dialog in the lower left) to move

the model down the Z-axis by -0.06096.





At this point You should have two collada files: `velodyne_base.dae` and `velodyne_top.dae` . These two files are also available from the following links:

1. `velodyne_base.dae`  
([https://github.com/osrf/gazebo\\_tutorials/raw/master/guided\\_i/files/velodyne\\_base.dae](https://github.com/osrf/gazebo_tutorials/raw/master/guided_i/files/velodyne_base.dae))
2. `velodyne_top.dae` ([https://github.com/osrf/gazebo\\_tutorials/raw/master/guided\\_i/files/velodyne\\_top.dae](https://github.com/osrf/gazebo_tutorials/raw/master/guided_i/files/velodyne_top.dae))

In the next section, we will cover adding these meshes to the SDF model.

## Step 2: Create the model structure

Gazebo has defined a model directory structure that supports stand-alone models, and the ability to share models via an online model database. Review this tutorial ([http://gazebosim.org/tutorials?tut=model\\_structure&cat=build\\_robot](http://gazebosim.org/tutorials?tut=model_structure&cat=build_robot)) for more information.

Another benefit of Gazebo's model structure is that it conveniently organizes resources, such as mesh files, required by the model. In this section, we will create a Velodyne SDF model and add the two mesh files, `velodyne_base.dae` and `velodyne_top.dae` .

1. Create a new directory to hold the Velodyne model. We will place the directory in `~/.gazebo/models` , since Gazebo knows to look there for models and this will speed the developement process.

```
mkdir ~/.gazebo/models/velodyne_hdl32
```

2. Create the `model.config` file. Each model requires some meta information that describes the model, the author, and any dependencies.

```
gedit ~/.gazebo/models/velodyne_hdl32/model.config
```

3. Copy the following into the `model.config` file.

```
<?xml version="1.0"?>

<model>
  <name>Velodyne HDL-32</name>
  <version>1.0</version>
  <sdf version="1.5">model.sdf</sdf>

  <author>
    <name>Optional: YOUR NAME</name>
    <email>Optional: YOUR EMAIL</email>
  </author>

  <description>
    A model of a Velodyne HDL-32 LiDAR sensor.
  </description>

</model>
```

4. Notice that the `model.config` file references a `model.sdf` file. This `model.sdf` file will contain the description of the Velodyne laser.

5. Create the `model.sdf` file.

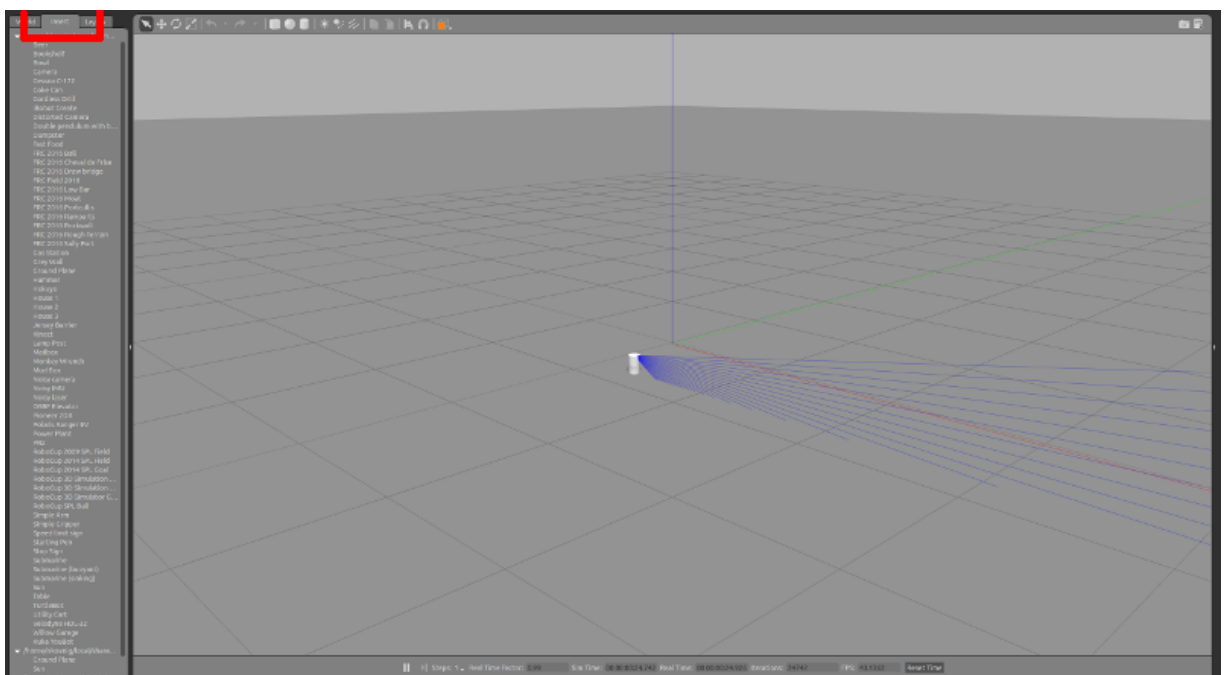
```
gedit ~/.gazebo/models/velodyne_hdl32/model.sdf
```

6. Copy the contents of `velodyne.world` into `model.sdf` and leave only the `<?xml>`, `<sdf>` and `<model>` tags, removing the following:

1. The opening and closing `<world>` tags
2. The `<include>` tags for sun and ground plane

7. At this point, we should be able to start Gazebo, and dynamically insert the Velodyne model.

1. gazebo
2. Select the **Insert** tab on the left, and scroll down to find the **Velodyne HDL-32** entry.
3. Click on the **Velodyne HDL-32** and then left-click in the render window to spawn the model.



# Step 3: Use the meshes

1. Create a `meshes` directory in the `~/gazebo/models/velodyne_hdl32` directory.

```
mkdir ~/.gazebo/models/velodyne_hdl32/meshes
```

2. Copy your two Collada files into the new directory.

```
cp velodyne_base.dae ~/.gazebo/models/velodyne_hdl32/meshes
cp velodyne_top.dae ~/.gazebo/models/velodyne_hdl32/meshes
```

3. Now we will modify the model's SDF to use the `velodyne_top` mesh.

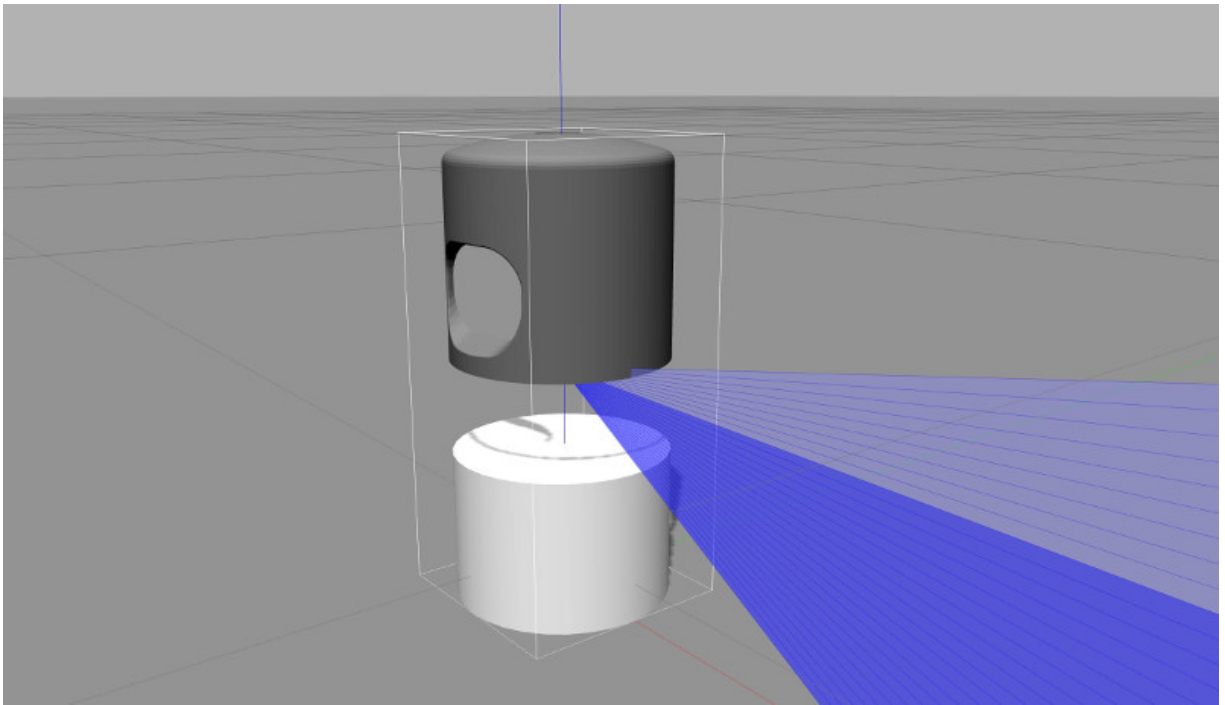
1. Open the `model.sdf` file:

```
gedit ~/.gazebo/models/velodyne_hdl32/model.sdf
```

2. Within the `<visual name="top_visual">` element, replace the `<cylinder>` element within with a `<mesh>` element. The `<mesh>` element should have a child `<uri>` that points to the top collada visual. The following snippet is what your top visual should contain.

```
<visual name="top_visual">
  <geometry>
    <!-- The mesh tag indicates that we will use a 3D mesh as
         a visual -->
    <mesh>
      <!-- The URI should refer to the 3D mesh. The "model:"
           URI scheme indicates that the we are referencing a Gazebo
           model. -->
      <uri>model://velodyne_hdl32/meshes/velodyne_top.dae</uri>
    </mesh>
  </geometry>
</visual>
```

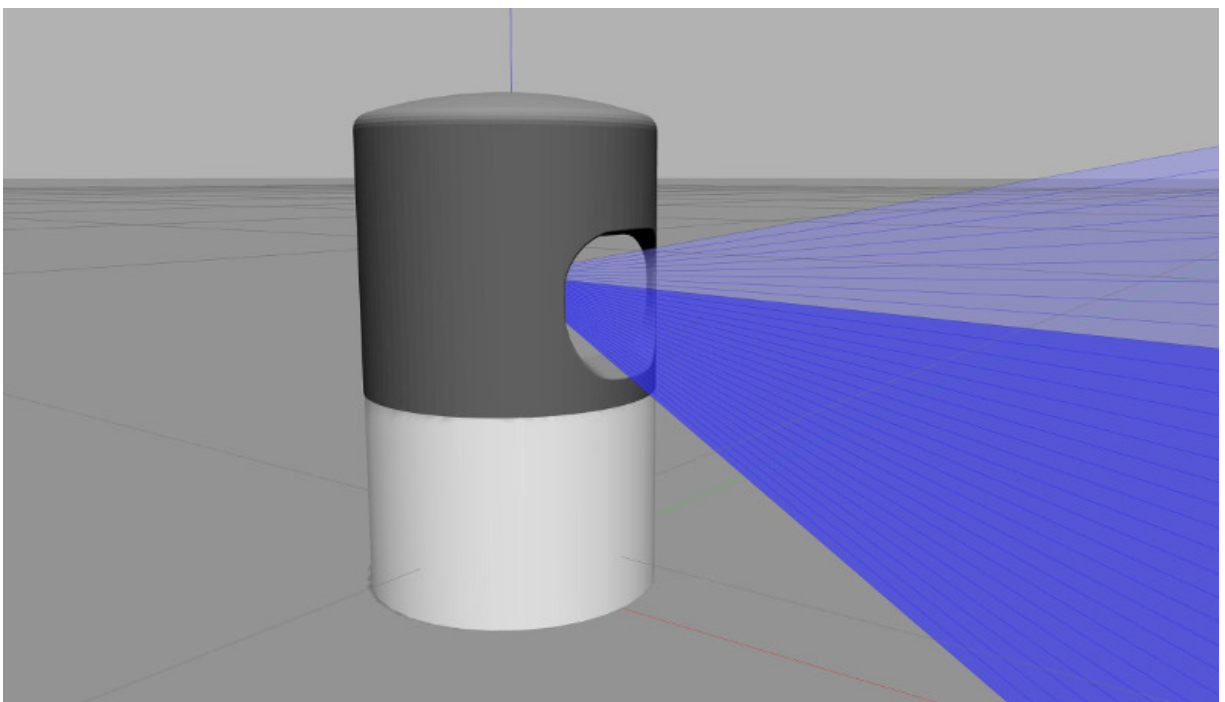
3. From Gazebo's **Insert** tab, insert another Velodyne HDL-32 model and you should see the following.



4. Notice that the visual is rotated incorrectly and has a vertical offset. These errors occur because the mesh's coordinate frame does not exactly match up with the coordinate frame of the SDF link. You can either edit the mesh in Blender to move the mesh, or you can apply a transform in SDF. Let's use the second option.

```
<visual name="top_visual">
  <!-- Lower the mesh by half the height, and rotate by 90 degrees -->
  <pose>0 0 -0.0376785 0 0 1.5707</pose>
  <geometry>
    <mesh>
      <uri>model://velodyne_hd132/meshes/velodyne_top.dae</uri>
    </mesh>
  </geometry>
</visual>
```

5. The result should now be correct.

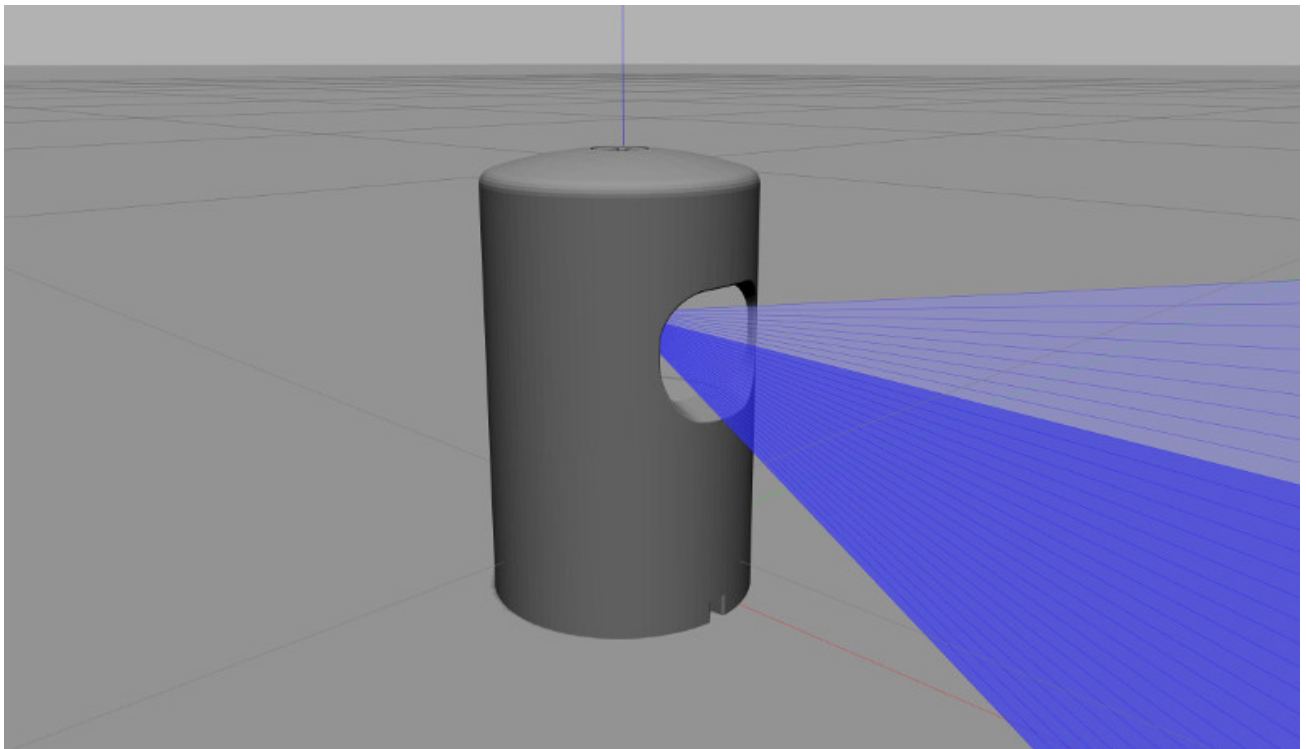




4. Now let's add the `velodyne_base` mesh, using what we learned from the `velodyne_top` mesh.

```
<visual name="base_visual">
  <!-- Offset the visual by have the base's height. We are not rotating
        mesh since symmetrical -->
  <pose>0 0 -0.029335 0 0 0</pose>
  <geometry>
    <mesh>
      <uri>model://velodyne_hd132/meshes/velodyne_base.dae</uri>
    </mesh>
  </geometry>
</visual>
```

5. We are done! The Velodyne model is looking good.



## Step 4: Textures

Textures add an additional level of realism. The Velodyne website does not have texture files for download, and for the most part the sensor is a uniform grey.

We will not add textures to the Velodyne model in this tutorial. However, if you have texture files then you can add them to your model in a couple ways.

1. Define a texture within a collada file, using texture mapping ([https://en.wikipedia.org/wiki/Texture\\_mapping](https://en.wikipedia.org/wiki/Texture_mapping)).
2. Define an OGRE material script ([http://www.ogre3d.org/docs/manual/manual\\_14.html](http://www.ogre3d.org/docs/manual/manual_14.html)), and attach it to the model using SDF ([http://sdformat.org/spec?ver=1.6&elem=material#material\\_script](http://sdformat.org/spec?ver=1.6&elem=material#material_script)).

## Next up

In the next tutorial we will add noise to the sensor reading. By default simulation will provide near-perfect data, and this does not match the data received in the real world.

Sensor Noise ([http://gazebo.org/tutorials?cat=guided\\_i&tut=guided\\_i3](http://gazebo.org/tutorials?cat=guided_i&tut=guided_i3))

©2014 Open Source Robotics  
Foundation

Gazebo is open-source licensed  
under Apache 2.0

(<http://www.apache.org/licenses/LICENSE-2.0.html>)

(<https://plus.google.com/u/0/11598143629prsrc=3>)

