

Conteúdo

1. The Marker Message

1. Example Usage (C++/roscpp)

2. Message Parameters

3. Object Types

1. Arrow (ARROW=0)

2. Cube (CUBE=1)

3. Sphere (SPHERE=2)

4. Cylinder (CYLINDER=3)

5. Line Strip (LINE_STRIP=4)

6. Line List (LINE_LIST=5)

7. Cube List (CUBE_LIST=6)

8. Sphere List (SPHERE_LIST=7)

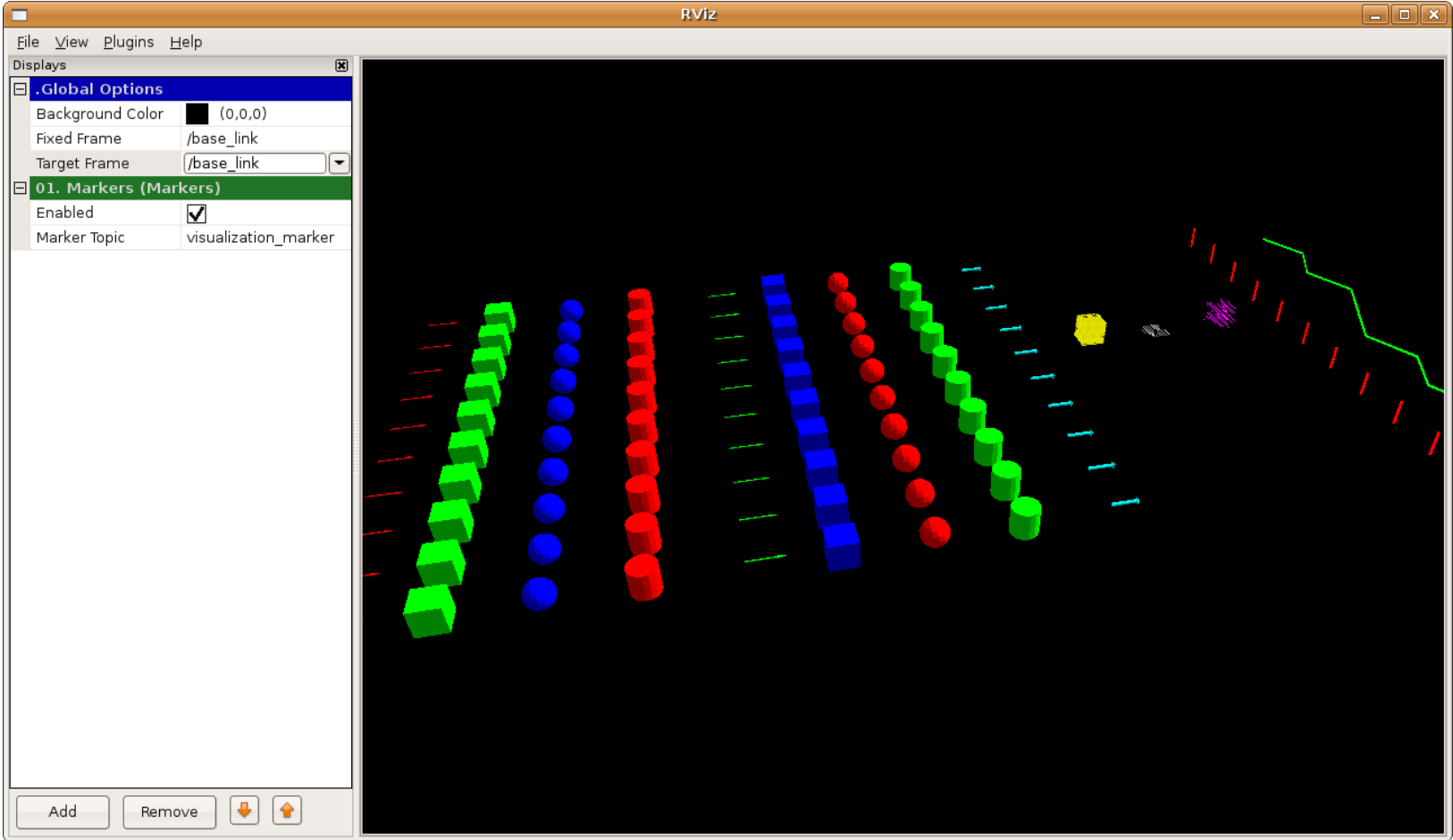
9. Points (POINTS=8)

10. View-Oriented Text (TEXT_VIEW_FACING=9) [1.1+]

11. Mesh Resource (MESH_RESOURCE=10) [1.1+]

12. Triangle List (TRIANGLE_LIST=11) [1.1+]

2. Rendering Complexity Notes



The **Markers** display allows programmatic addition of various primitive shapes to the 3D view by sending a visualization_msgs/Marker (http://docs.ros.org/api/visualization_msgs/html/msg/Marker.html) or visualization_msgs/MarkerArray (http://docs.ros.org/api/visualization_msgs/html/msg/MarkerArray.html) message.

The Markers: Basic Shapes (</rviz/Tutorials/Markers%3A%20Basic%20Shapes>) tutorial begins a series of tutorials on sending markers.

Properties

Name	Description	Valid Values	Default
Topic	The topic to subscribe to. Will automatically also subscribe to <topic>_array, assuming it is a visualization_msgs/MarkerArray (http://docs.ros.org/api/visualization_msgs/html/msg/MarkerArray.html) topic	Any valid Graph Resource Name (/Names#Graph)	visualization_marker

1. The Marker Message

1.1 Example Usage (C++/roscpp)

First, advertise on the visualization_marker topic:

Ligar/Desligar números de linhas

```
1 ros::Publisher vis_pub = node_handle.advertise<visualization_msgs::Marker>( "visualization_marker",
0 );
```

After that it's as simple as filling out a visualization_msgs/Marker (http://docs.ros.org/api/visualization_msgs/html/msg/Marker.html) message and publishing it:

Ligar/Desligar números de linhas

```
1 visualization_msgs::Marker marker;
2 marker.header.frame_id = "base_link";
3 marker.header.stamp = ros::Time();
4 marker.ns = "my_namespace";
5 marker.id = 0;
6 marker.type = visualization_msgs::Marker::SPHERE;
7 marker.action = visualization_msgs::Marker::ADD;
8 marker.pose.position.x = 1;
9 marker.pose.position.y = 1;
10 marker.pose.position.z = 1;
11 marker.pose.orientation.x = 0.0;
12 marker.pose.orientation.y = 0.0;
13 marker.pose.orientation.z = 0.0;
14 marker.pose.orientation.w = 1.0;
15 marker.scale.x = 1;
16 marker.scale.y = 0.1;
17 marker.scale.z = 0.1;
18 marker.color.a = 1.0;
19 marker.color.r = 0.0;
20 marker.color.g = 1.0;
21 marker.color.b = 0.0;
22 //only if using a MESH_RESOURCE marker type:
23 marker.mesh_resource = "package://pr2_description/meshes/base_v0/base.dae";
24 vis_pub.publish( marker );
```

There is also a visualization_msgs/MarkerArray (http://docs.ros.org/api/visualization_msgs/html/msg/MarkerArray.html) message, which lets you publish many markers at once. In this case you want to publish in the visualization_marker_array topic.

Note that the timestamp attached to the marker message above is ros::Time(), which is time Zero (0). This is treated differently by RViz than any other time. If you use ros::Time::now() or any other non-zero value, rviz will only display the marker if that time is close enough to the current time, where "close enough" depends on TF. With time 0 however, the marker will be displayed regardless of the current time.

1.2 Message Parameters

ns	Namespace for these markers. This plus the id form a unique identifier.
id	Unique id assigned to this marker. It is your responsibility to keep these unique within your namespace.
type	Type of marker (Arrow, Sphere, ...). The available types are specified in the message definition.
action	0 = add/modify, 1 = (deprecated), 2 = delete, New in Indigo 3 = deleteall
pose	Pose marker, specified as x/y/z position and x/y/z/w quaternion orientation.
scale	Scale of the marker. Applied before the position/orientation. A scale of [1,1,1] means the object will be 1m by 1m by 1m.
color	Color of the object, specified as r/g/b/a, with values in the range of [0, 1]

points

Only used for markers of type Points, Line strips, and Line/Cube/Sphere lists.

colors [1.1+]

Only used for markers that use the points member, specifies per-vertex color (no alpha yet).

lifetime

A duration value used to automatically delete the marker after this period of time. The countdown resets if another marker of the same namespace/id is received.

frame_locked [1.1+]

Tells rviz to retransform the marker into the current location of the specified frame every update cycle.

text [1.1+]

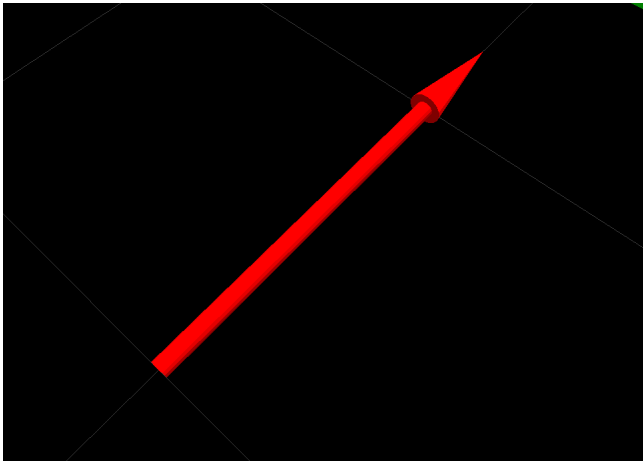
The text string used for the TEXT_VIEW_FACING marker type

mesh_resource [1.1+]

The resource location for the MESH_RESOURCE marker type. Can be any mesh type supported by rviz (.stl or Ogre .mesh in 1.0, with the addition of COLLADA in 1.1). The format is the URI-form used by resource_retriever (/resource_retriever), including the package:// syntax.

1.3 Object Types

1.3.1 Arrow (ARROW=0)



The arrow type provides two different ways of specifying where the arrow should begin/end:

Position/Orientation

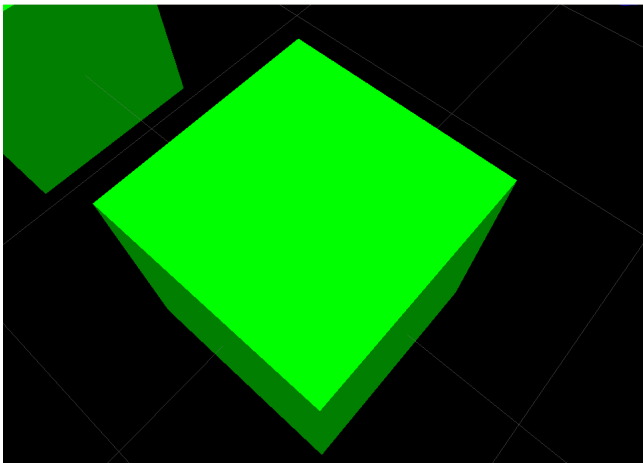
Pivot point is around the tip of its tail. Identity orientation points it along the +X axis. `scale.x` is the arrow length, `scale.y` is the arrow width and `scale.z` is the arrow height.

Start/End Points

You can also specify a start/end point for the arrow, using the points member. If you put points into the points member, it will assume you want to do things this way.

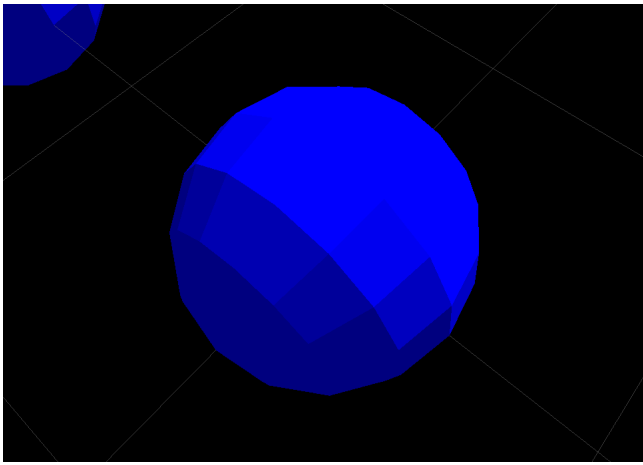
- The point at index 0 is assumed to be the start point, and the point at index 1 is assumed to be the end.
- `scale.x` is the shaft diameter, and `scale.y` is the head diameter. If `scale.z` is not zero, it specifies the head length.

1.3.2 Cube (CUBE=1)



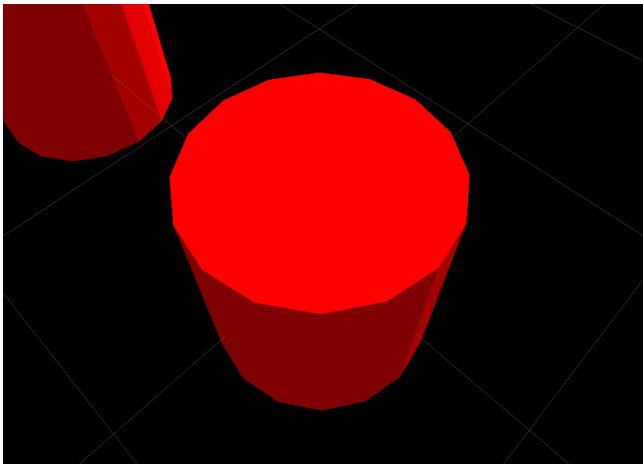
Pivot point is at the center of the cube.

1.3.3 Sphere (SPHERE=2)



Pivot point is at the center of the sphere.

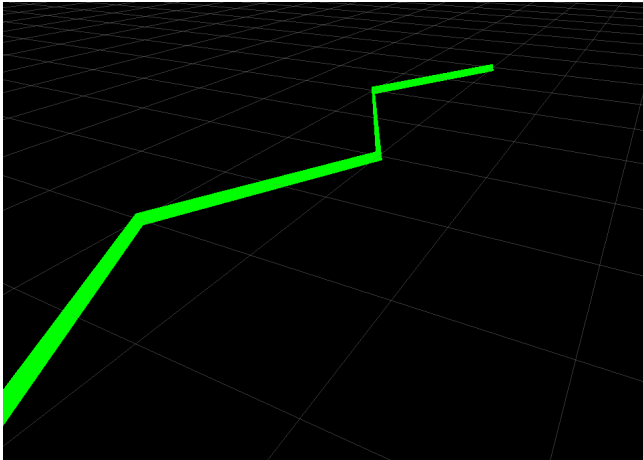
1.3.4 Cylinder (CYLINDER=3)



Pivot point is at the center of the cylinder.

`scale.x` is diameter in x direction, `scale.y` in y direction, by setting these to different values you get an ellipse instead of a circle. Use `scale.z` to specify the height.

1.3.5 Line Strip (LINE_STRIP=4)



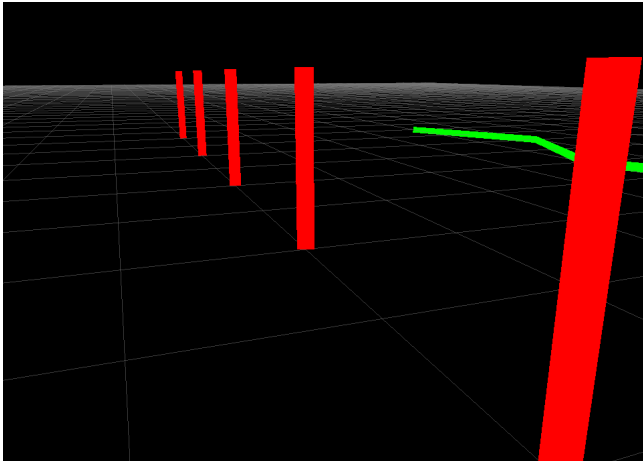
Line strips use the `points` member of the `visualization_msgs/Marker` (http://docs.ros.org/api/visualization_msgs/html/msg/Marker.html) message. It will draw a line between every two consecutive points, so 0-1, 1-2, 2-3, 3-4, 4-5...

Line strips also have some special handling for `scale`: **only `scale.x` is used** and it controls the width of the line segments.

Note that `pose` is still used (the points in the line will be transformed by them), and the lines will be correct relative to the frame id specified in the header.

In visualization 1.1+ will also optionally use the `colors` member for per-vertex color.

1.3.6 Line List (LINE_LIST=5)



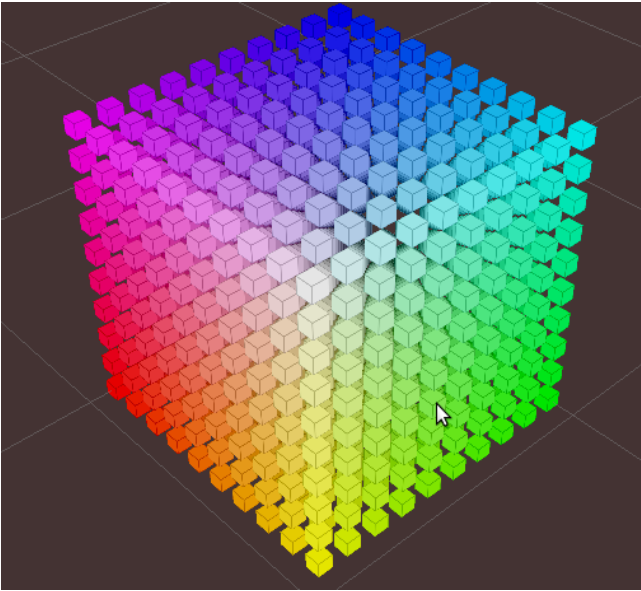
Line lists use the `points` member of the `visualization_msgs/Marker` (http://docs.ros.org/api/visualization_msgs/html/msg/Marker.html) message. It will draw a line between each pair of points, so 0-1, 2-3, 4-5, ...

Line lists also have some special handling for `scale`: **only `scale.x` is used** and it controls the width of the line segments.

Note that `pose` is still used (the points in the line will be transformed by them), and the lines will be correct relative to the frame id specified in the header.

In visualization 1.1+ will also optionally use the colors member for per-vertex color.

1.3.7 Cube List (CUBE_LIST=6)

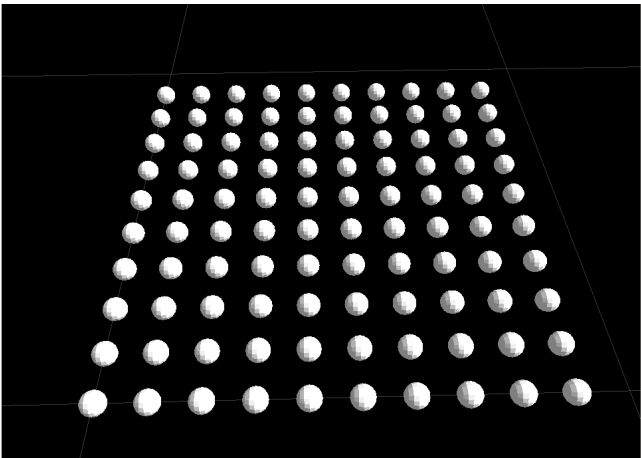


A cube list is a list of cubes with all the same properties except their positions. Using this object type instead of a visualization_msgs/MarkerArray (http://docs.ros.org/api/visualization_msgs/html/msg/MarkerArray.html) allows rviz to batch-up rendering, which causes them to render much faster. The caveat is that they all must have the same color/scale.

The points member of the visualization_msgs/Marker (http://docs.ros.org/api/visualization_msgs/html/msg/Marker.html) message is used for the position of each cube.

In visualization 1.1+ will also optionally use the colors member for per-cube color.

1.3.8 Sphere List (SPHERE_LIST=7)

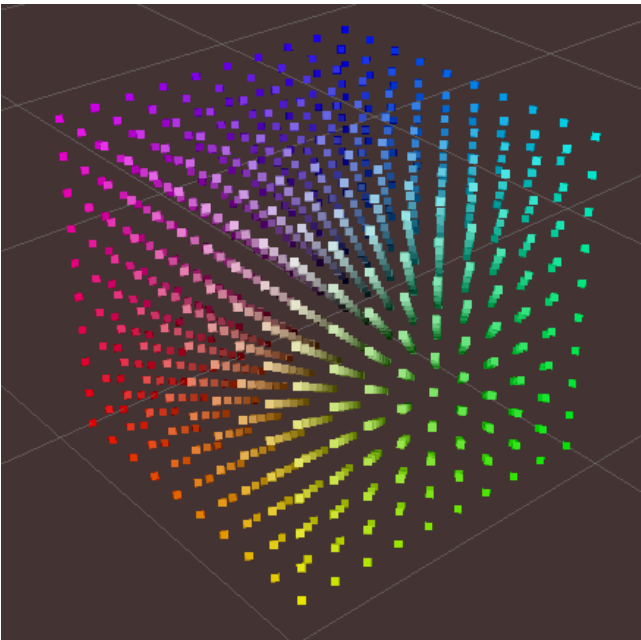


A sphere list is a list of spheres with all the same properties except their positions. Using this object type instead of a visualization_msgs/MarkerArray (http://docs.ros.org/api/visualization_msgs/html/msg/MarkerArray.html) allows rviz to batch-up rendering, which causes them to render much faster. The caveat is that they all must have the same color/scale.

The points member of the visualization_msgs/Marker (http://docs.ros.org/api/visualization_msgs/html/msg/Marker.html) message is used for the position of each sphere.

Note that pose is still used (the points in the line will be transformed by them), and the lines will be correct relative to the frame id specified in the header.

1.3.9 Points (POINTS=8)



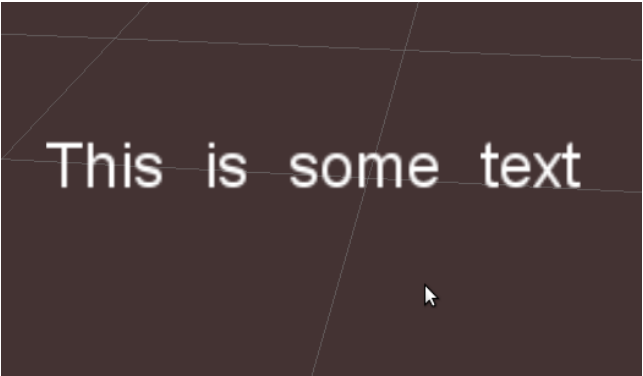
Uses the points member of the visualization_msgs/Marker (http://docs.ros.org/api/visualization_msgs/html/msg/Marker.html) message.

Points have some special handling for scale: **scale.x is point width, scale.y is point height**

Note that pose is still used (the points in the line will be transformed by them), and the lines will be correct relative to the frame id specified in the header.

In visualization 1.1+ will also use the colors member for per-point color.

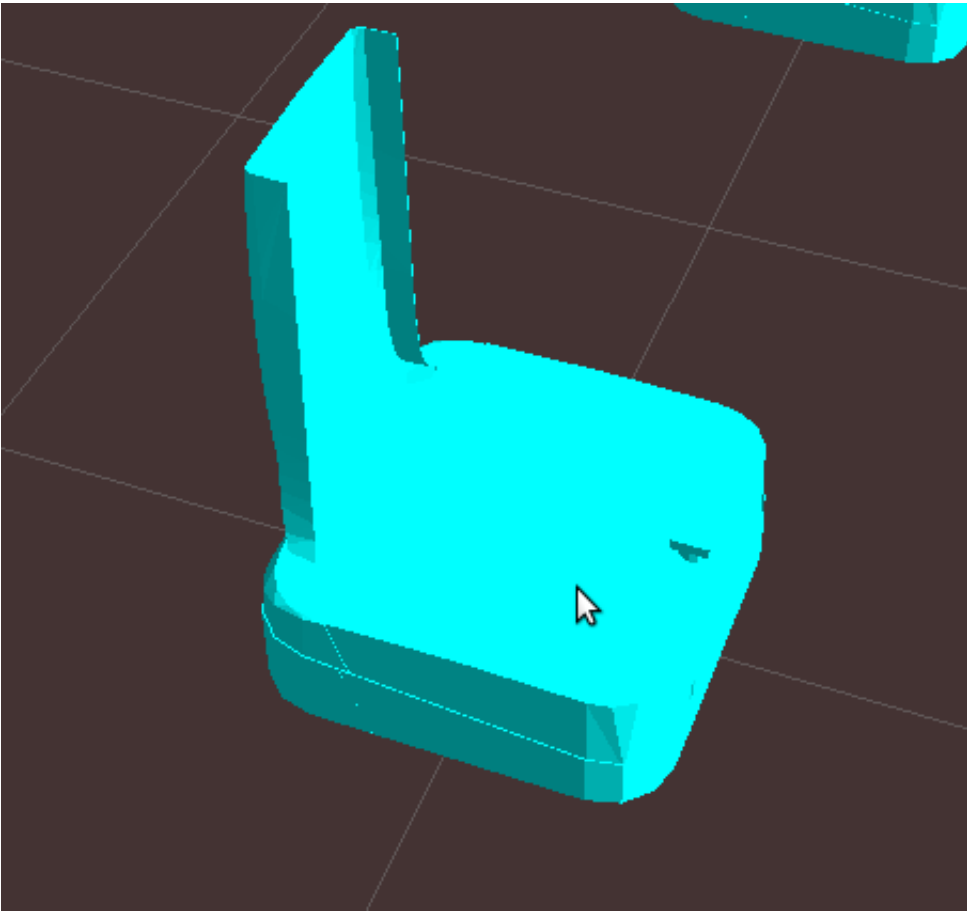
1.3.10 View-Oriented Text (TEXT_VIEW_FACING=9) [1.1+]



This marker displays text in a 3D spot in the world. The text always appears oriented correctly to the view. Uses the text field in the marker.

Only scale.z is used. scale.z specifies the height of an uppercase "A".

1.3.11 Mesh Resource (MESH_RESOURCE=10) [1.1+]



Uses the mesh_resource field in the marker. Can be any mesh type supported by rviz (binary .stl or Ogre .mesh in 1.0, with the addition of COLLADA (.dae) in 1.1). The format is the URI-form used by resource_retriever (/resource_retriever), including the package:// syntax.

An example of a mesh an its use is

```
Ligar/Desligar números de linhas

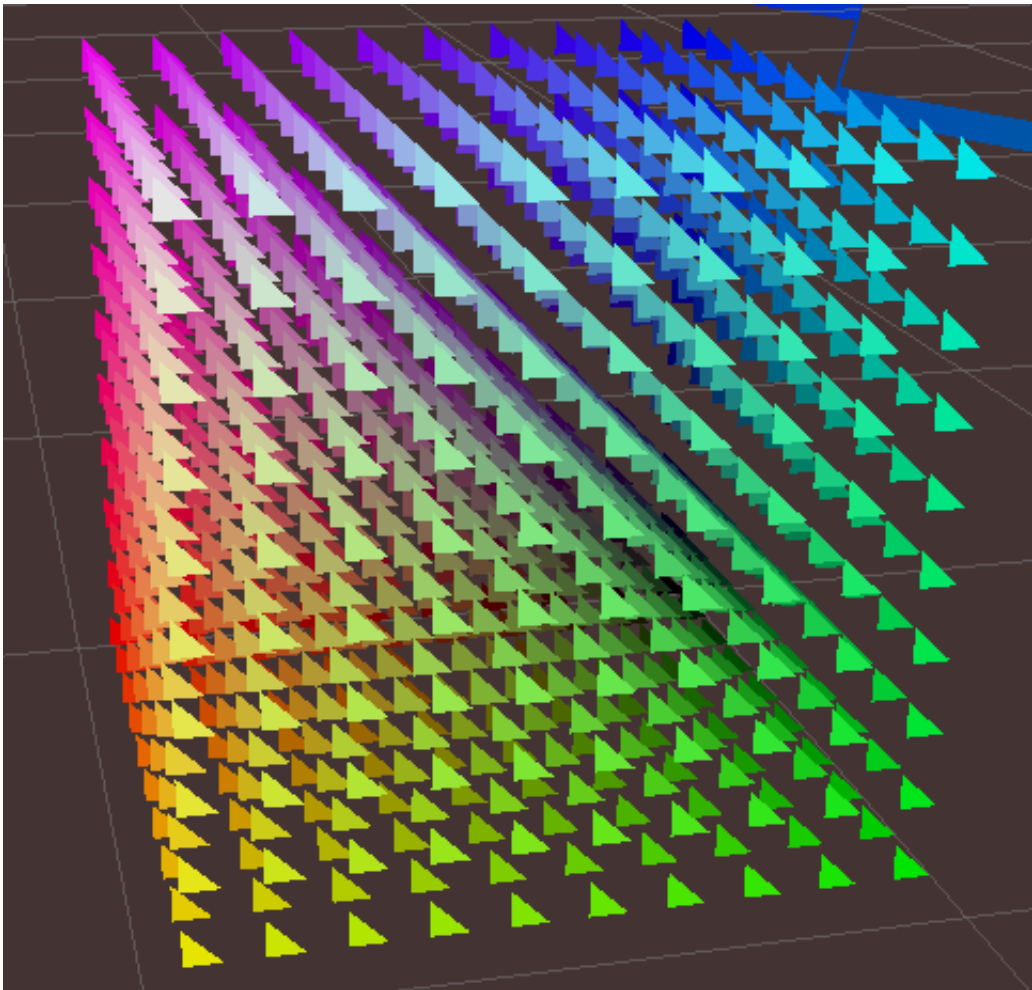
1 marker.type = visualization_msgs::Marker::MESH_RESOURCE;
2 marker.mesh_resource = "package://pr2_description/meshes/base_v0/base.dae";
```

Scale on a mesh is relative. A scale of (1.0, 1.0, 1.0) means the mesh will display as the exact size specified in the mesh file. A scale of (1.0, 1.0, 2.0) means the mesh will show up twice as tall, but the same width/depth.

If the mesh_use_embedded_materials flag is set to true and the mesh is of a type which supports embedded materials (such as COLLADA), the material defined in that file will be used instead of the color defined in the marker.

Since version **[1.8]**, even when mesh_use_embedded_materials is true, if the marker color is set to anything other than r=0,g=0,b=0,a=0 the marker color and alpha will be used to tint the mesh with the embedded material.

1.3.12 Triangle List (TRIANGLE_LIST=11) [1.1+]



Uses the `points` and optionally `colors` members. Every set of 3 points is treated as a triangle, so indices 0-1-2, 3-4-5, etc.

Note that `pose` and `scale` are still used (the points in the line will be transformed by them), and the lines will be correct relative to the frame id specified in the header.

2. Rendering Complexity Notes

A single marker is always less expensive to render than many markers. For example, a single cube list can handle thousands of cubes, where we will not be able to render thousands of individual cube markers.

Except where otherwise noted, the ROS wiki is licensed under the Wiki: rviz/DisplayTypes/Marker (última edição 2014-06-18 23:56:01 efectuada por davetcoleman (/davetcoleman))
Creative Commons Attribution 3.0
(<http://creativecommons.org/licenses/by/3.0/>) | Find us on Google+ (<https://plus.google.com/113789706402978299308>)

Brought to you by:  Open Source Robotics Foundation

(<http://www.osrfoundation.org>)