

00A0 251C+2514+

---

# **jsk\_visualization Documentation**

***Release 2.1.7***

**Ryohei Ueda, Kei Okada, Youhei Kakiuchi**

**Dec 13, 2021**



# Contents

<b>1</b>	<b>jsk_rviz_plugins</b>	<b>3</b>
<b>2</b>	<b>jsk_rqt_plugins</b>	<b>35</b>
<b>3</b>	<b>jsk_interactive_marker</b>	<b>45</b>
<b>4</b>	<b>Tips about visualization</b>	<b>59</b>



jsk\_visualization is a stack for the visualization packages which are used in JSK lab.

The code is open source, and available on [github](#).

This repository contains following ros packages:



# Chapter 1

## jsk\_rviz\_plugins

jsk\_rviz\_plugins is a package to provide original rviz plugins.

You can use these rviz plugins, panels and tools just by launching rviz.

### 1.1 SegmentArray

#### 1.1.1 What is this?

Visualize `jsk_recognition_msgs/SegmentArray.msg`.

### **1.1.2 Sample**



## **1.2 BoundingBox**

### **1.2.1 What is this?**

Visualize `jsk_recognition_msgs/BoundingBox.msg`.

## **1.3 BoundingBoxArray**

### **1.3.1 What is this?**

Visualize `jsk_recognition_msgs/BoundingBoxArray.msg`.

## 1.4 CameraInfo

Visualize sensor\_msgs/CameraInfo.



## 1.5 Footstep

### 1.5.1 What is this?

Visualize `jsk_footstep_msgs/Footstep.msg`.

## **1.6 HumanSkeltonArray**

Visualize jsk\_recognition\_msgs/HumanSkeltonArray

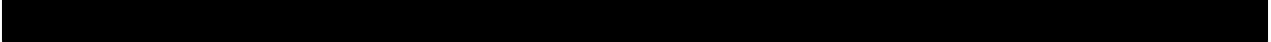
## **1.7 LinearGauge**

Plot a linear gauge of std\_msgs/Float32 on rviz as HUD overlay.

To change caption text, please rename plugin display name on rviz Displays tab

It can increase in either horizontal or vertical direction

### **1.7.1 Sample**



## **1.8 Normal**

This will show the Normal which is subscribed from topic (sensor\_msgs::PointCloud2). The normal is assumed to have the features x,y,z,normal\_x,normal\_y,normal\_z.

Normal  
Plugin

### **1.8.1 Samples**



Plug the depth sensor which could be launched by openni.launch and execute below command.

## **1.9 OverlayText**

Draw text of jsk\_rviz\_plugins/OverlayText on rviz as HUD overlay.

### **1.9.1 Sample**



## **1.10 PeoplePositionMeasuermentArray**

It visualizes `people_msgs/PositionMeasuermentArray`.

## **1.11 Pictogram**

Pictogram

movie

Pictogram is a rviz plugin to visualize icons. Pictogram plugin uses Entypo and FontAwesome.

You need to use `jsk_rviz_plugins/Pictogram` and `jsk_rviz_plugins/PictogramArray` message to use it.

You can find mapping with `character` and icons at [here](#) and [here](#).

If you set `STRING_MODE`, you can show the string popups.

## 1.12 PieChart

Plot a pie chart of std\_msgs/Float32 on rviz as HUD overlay.

To change caption text, please rename plugin display name on rviz Displays tab

### 1.12.1 Properties

- Topic  
std\_msgs::Float32 topic to subscribe to
- size  
Size of the plotter window
- left  
Left of the plotter window
- top  
Top of the plotter window
- foreground color  
Color to draw line
- foreground alpha  
Alpha belnding value for foreground
- foreground alpha2  
Alpha belnding value for foreground for indicator
- background color  
Background color
- background alpha  
Alpha belnding value for background

- text size  
Text size
- show caption  
Show caption
- max value  
Max value of pie chart
- min value  
Min value of pie chart
- auto color change  
Change the color automatically
- max color  
Max color of pie chart  
Only used if auto color change is set to True
- med color  
Med color of pie chart  
Only used if auto color change is set to True
- max color change threshold  
Change the max color at this threshold  
Only used if auto color change is set to True
- med color change threshold  
Change the med color at this threshold  
Only used if auto color change is set to True
- clockwise rotate direction  
Change the rotate direction

### 1.12.2 Sample

[REDACTED]

or

[REDACTED]

## **1.13 Plotter2D**

Plot a line graph of std\_msgs/Float32 on rviz as HUD Display.

To change caption text, please rename plugin display name on rviz Displays tab

### **1.13.1 Sample**

## 1.14 PolygonArray

PolygonArray

Visualize `jsk_recognition_msgs/PolygonArray` message

### 1.14.1 Properties

- Topic

Name of topic of `jsk_recognition_msgs/PolygonArray`

- auto\_color

If it's true, color of polygons are automatically changed

- Color

Color of polygons, only enabled if `auto_color` is false

- Alpha

Transparency of polygons

- only\_border

Draws only edges of polygons.

- show\_normal

Show normal of polygons.

- nromal\_length

Lenght of normal [m].

## 1.15 RvizScenePublisher

### 1.15.1 What is this?

RvizScenePublisher plugin can publish `sensor_msgs/Image` of rviz.

### 1.15.2 Publishing Topic

- `/rviz/image` (`sensor_msgs/Image`)

Scene of rviz image. You can change the topic name by changing `topic_name` in Displays.

## 1.16 SimpleOccupancyGridArray

Visualize jsk\_recognition\_msgs/SimpleOccupancyGridArray.

## 1.17 String

Draw text of std\_msgs/String on rviz as HUD overlay.

### **1.17.1 Sample**



## 1.18 TFTrajectory

Visualize trajectory of a tf frame.

<https://youtu.be/d9>

## 1.19 TorusArray

TorusArray

Visualize `jsk_pcl_ros/TorusArray` message

### 1.19.1 Properties

- Topic  
Name of topic of `jsk_pcl_ros/TorusArray`
- auto\_color  
If it's true, color of polygons are automatically changed
- Color  
Color of polygons, only enabled if `auto_color` is false
- Alpha  
Transparency of polygons
- uv-smooth  
Smoothness the surface
- show\_normal  
Show normal of toruses.
- nromal\_length  
Length of normal [m].

## 1.20 TwistStamped

Movie

Visualize geometry\_msgs/TwistStamped by arrows. Linear velocity is represented by one arrow and angular velocity is represented by 3 arrows for each axis.

### 1.20.1 Properties

- linear scale (default: 1.0)
- angular scale (default: 1.0)

Scale factor of size of arrows

- linear color (default: RGB(0, 255, 0))
- angular color (default: RGB(255, 0, 0))

Color of arrows

## **1.21 VideoCapture**

VideoCapture plugin can capture video of rviz.

You need to specify valid filename and fps before capturing video. You can also specify width and height of video manually instead of using 3D viewer size if you want. After that, toggle start capture checkbox and the movie will be recorded until you uncheck the checkbox.

Be careful on creating too large video.

## **1.22 CancelAction**

CancelAction

This will publish action\_msg/GoalID to topic\_name/cancel. You can choose multiple cancel goals.

## 1.23 ObjectFitOperatorAction

ObjectFitOperatorAction

This will publish jsk\_rviz\_plugins/ObjectFitCommand to /object\_fit\_command. If you check reversed, the reversed version will publish.

## 1.24 PublishTopic

PublishTopic

This will publish std\_msgs/Empty to the topic you designate.

## 1.25 RecordAction

RecordAction

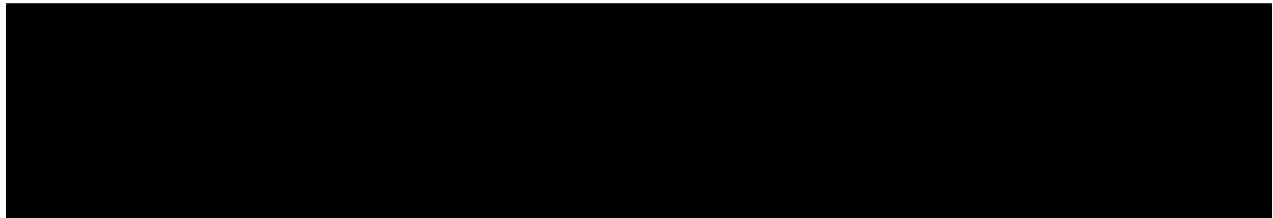
This will publish jsk\_rviz\_plugins/RecordCommand to /record\_command. Set the target name.

## 1.26 RobotCommandInterfaceAction

RobotCommandIn

This will call service to /eus\_command with jsk\_rviz\_plugins/EusCommand srv. All the buttons are configured via ~robot\_command\_buttons parameters. See `robot_command_interface_sample.launch` file to know how to use it.

Parameter format is:



## 1.27 SelectPointCloudPublishAction

SelectPointCloudPublishAction

This will publish sensor\_msgs/PointCloud2 to /selected\_pointcloud.

1. First, push SelectButton and select the pointcloud region(Note that you need to choose only pointcloud. Don't include other parts).
2. Secord, push the SelectPointCloudPublishAction button.

3. Then the selected pointcloud will be published.

## **1.28 TransformableMarkerOperatorAction**

TransformableMarkerOperatorAction

TransformableMarkerOperatorAction

This will call service to /request\_marker\_operate to insert/erase transformable\_object

## 1.29 YesNoButton

Get yes/no user input with rviz button interface.

### 1.29.1 Advertising Services

- /rviz/yes\_no\_button (jsk\_gui\_msgs/YesNo)  
Service server to get yes/no user input.

## 1.30 OverlayPickerTool

It is a tool to move overlay plugins interactively.

If you drag overlay widget with pressing Shift key, the widget is aligned to grid.

## 1.31 ScreenshotListenerTool

It is a tool to take screenshot of rviz via service interface.

Click ↞Jadd button⁞ of toolbar on rviz and you will see popup to add tools.



And /rviz/screenshot service will be available.

You can save screenshot via service call:

## **1.32 classification\_result\_visualizer.py**

### **1.32.1 What is this?**

Publish classification results as text markers for each classified object.

The ClassificationResult is synchronized with one topic which represents poses for each classified object / person.

### 1.32.2 Subscribing Topic

- ~input/classes (jsk\_recognition\_msgs/ClassificationResult)  
Classification result
- ~input/boxes (jsk\_recognition\_msgs/BoundingBoxArray)  
Bounding boxes of classified objects
- ~input/poses (geometry\_msgs/PoseArray)  
Poses of classified objects
- ~input/people (jsk\_recognition\_msgs/PeoplePoseArray)  
Poses of classified people
- ~input/ObjectDetection (posedetection\_msgs/ObjectDetection)  
Poses of classified objects

### 1.32.3 Publishing Topic

- ~output (visualization\_msgs/MarkerArray)  
Text message markers

### 1.32.4 Parameters

- ~approximate\_sync (Bool, Default: false)  
Option to enable approximate synchronization
- ~queue\_size (Int, Default: 100)  
Queue size of subscribers on synchronization
- ~slop (Double, Default: 0.1)  
Slop duration on approximate synchronization
- ~text\_color\_blue (Double, Default: 1.0)  
Blue of text color
- ~text\_color\_green (Double, Default: 0.0)  
Green of text color
- ~text\_color\_red (Double, Default: 0.0)  
Red of text color
- ~text\_color\_alpha (Double, Default: 1.0)  
Alpha of text color
- ~text\_offset\_x (Double, Default: 0.0)  
Text offset on x-axis for each object
- ~text\_offset\_y (Double, Default: 0.0)  
Text offset on y-axis for each object

- `~text_offset_z` (Double, Default: 0.07)  
Text offset on y-axis for each object
- `~text_size` (Double, Default: 0.05)  
Text size
- `~marker_lifetime` (Double, Default: 5.0)  
Marker lifetime
- `~show_proba` (Bool, Default: true)  
Enable to display probability for each classification

## **1.33 rosconsole\_overlay\_text.py**

### **1.33.1 What is this?**

Publish message for overlaying ROS console output on rviz.

### 1.33.2 Subscribing Topic

- `/rosout` (`rosgraph_msgs/Log`)

ROS console output.

### 1.33.3 Publishing Topic

- `~output` (`jsk_rviz_plugins/OverlayText`)

Text message displayed on rviz with OverlayText plugin.

### 1.33.4 Parameters

- `~nodes` (List of String, Default: `[]`)

Node names whose messages will be published. If an empty list is specified (default), then messages from all nodes will be published.

- `~nodes_regex` (String, Default: `" "`)

Regular expression used to filter unmatching nodes. If an empty string is specified (default), then messages from all nodes will be published.

Note that the combination of `~nodes` and `~nodes_regex` is AND filter.

- `~ignore_nodes` (List of String, Default: `[]`)

Node names whose messages won't be published. This parameter takes priority over `~nodes` or `nodes_regex`.

- `~exclude_regexes` (List of String, Default: `[]`)

Regular expressions used to exclude matching messages.

- `~line_buffer_length` (Int, Default: 100)

Max number of messages stored in buffer.

- `~reverse_lines` (Bool, default: True)

If True, the order of stored messages in the buffer will be reversed.

### 1.33.5 Sample



# **Chapter 2**

## **jsk\_rqt\_plugins**

rqt\_plugins created in JSK Lab.

### **2.1 rqt\_2d\_plot**

Plot data of specified topic as scatter plot.

### **2.1.1 Topic Type**

- `jsk_recognition_msgs/PlotData`

### **2.1.2 Optional Arguments**

- `--line`: Plot with lines instead of scatter.
- `--fit-line`: Plot line with least-square fitting.
- `--fit-line-ransac`: Plot line with RANSAC.
- `--fit-line-ransac-outlier`: Plot line with RANSAC.

### **2.1.3 Sample**



## **2.2 rqt\_3d\_plot**

Plot multiple topics in 3-dimentional layout.

### **2.2.1 Topic Type**

- numeric data such as `std_msgs/Float32`

## 2.2.2 Optional Arguments

- -P, --pause: Start in paused state.
- -L, --line: Show lines rather than polygon representation.
- --no-legend: Do not show legend.
- -B, --buffer: The length of the buffer. (default = 100)

## 2.2.3 Sample



## 2.3 rqt\_drc\_mini\_maxwell

Subscribe specified topic and show status in facial expression.

### 2.3.1 Subscribing Topic

- /drc\_2015\_environment/is\_disabled(std\_msgs/Bool)
- /drc\_2015\_environment/is\_blackout(std\_msgs/Bool)
- /drc\_2015\_environment/next\_whiteout\_time(std\_msgs/Time)

If `is_disabled` is True, then it frowns and the background color becomes gray.

If `is_disabled` is False and `is_blackout` is True, then it frowns and the background color becomes red. `next_whiteout_time` is enabled only in this condition.

If `is_disabled` is False and `is_blackout` is False, then it smiles and the background color becomes green.

### 2.3.2 Sample



## **2.4 rqt\_histogram\_plot**

Plot histogram data. It supported array fields of topics and `jsk_recognition_msgs/HistogramWithRange`. If you want to specify x-values of figure, use `jsk_recognition_msgs/HistogramWithRange`.

## **2.5 rqt\_image\_view2**

rqt wrapper of `image_view2`.

It retrieves `image_marked` topic from `image_view2` and publish event topic to `image_view2`.

### **2.5.1 Sample**

## 2.6 rqt\_service\_buttons

Generate service buttons according to the configuration written in yaml file.

- sample yaml file: `jsk_rqt_plugins/resource/service_button_layout.yaml`
- sample perspective file: `jsk_rqt_plugins/resource/rqt_service_buttons.perspective`

### 2.6.1 Sample



## **2.7 rqt\_service\_buttons**

Generate service buttons according to the configuration written in yaml file.

- sample yaml file: `jsk_rqt_plugins/resource/service_radio_button_layout.yaml`
- sample perspective file: `jsk_rqt_plugins/resource/rqt_service_radio_buttons.perspective`

### **2.7.1 Sample**



## 2.8 rqt\_status\_light

Subscribe specified topic and show status in a simple color.

### 2.8.1 Topic Type

- std\_msgs/UInt8

### 2.8.2 Correspondence between Value and Color

- 1: green
- 2: yellow
- (other): gray

### 2.8.3 Usage



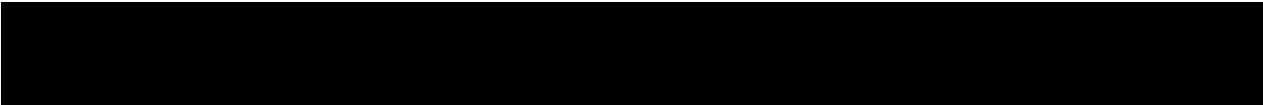
## **2.9 rqt\_string\_label**

Subscribe string under specified topic and show the string message.

### **2.9.1 Topic Type**

- Any topic type which has `string` slot

### **2.9.2 Usage**



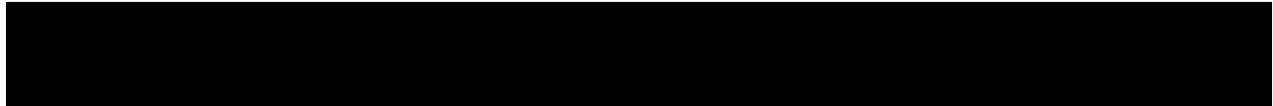
## **2.10 rqt\_yn\_btn**

Serves yes/no buttons. The buttons are enabled when there is a request.

### **2.10.1 Advertising Service**

- `rqt_yn_btn (jsk_rqt_plugins/YesNo)`

### **2.10.2 Usage**





## Chapter 3

### jsk\_interactive\_marker

jsk\_interactive\_marker is a code using interactive marker.

#### 3.1 moveit\_msgs/DisplayRobotState ČSåLı'çTíäAžäAervizäyLäAñäČ■äČIjä

rvižäyLäAñèGlıçTıäAñäČ■äČIjäČCäČLäČSéE■ç; öäAÜäAäRfreÜäNÜäAžäČNäAñäAmoveit\_msgs/  
DisplayRobotState äAÑä; £äLı'äAğäAžäAČ äAŞäČNäČSåLı'çTíäAžäČNäAŞäAíäAğäAäéÜççíAègŠåž+äCñäČijäČLäČlaČşäČ  
äČäČşäČÜäČñäAfrroslaunch jsk\_interactive\_marker sample\_display\_robot\_state.  
launchäAğççž■äAğäA■äAğäAžäAČ  
moveit\_msgs/DisplayRobotState äČSåLı'çTíäAžäČNäAŞäČAäAñäAfräAmoveit\_ros\_visualizationäAñäČL'Rob



ãAŞaAőRobotStateãČÜaČl'ãČrāC'd'ãČşaař2aAđ'ãAđaČS'aČl'ãČqaaČijaaČfaaĀA1aAđ'ãAđaČL'aČT'aČC'aČrāAňač'lařYaaUaAč'aAžaĀC

- parameter

- /robot\_description

ãČiaČIjaaČC'aČL'aAđač'lařS'eřaČAaAšaaČAaAňaĀArobot\_descriptionãČS'ařEeřAaAļaAđaAč'aAžaĀC

- /robot\_description\_semantic

moveitAňgāL'cTílaAřaČN'aĀAđaČd'aČN'aČT'aČeāČijaaČfæČEňašçL'aČS'ařiřaČN'aĀCš'aČl'ãČqaaČijaaČfaaĀC  
ažořČşaaČyaaČgaaČ'd'aČşaaČL'aČS'eř; aL'aAžaĀC'N'aAšaaĀAňaAđaAč'aAšaaČAaĀAňaAšaaČN'aĀCš'aL'cTílaAđaAęaČn'aĀC

- topic

- Robot State Topic (moveit\_msgs/DisplayRobotState)

ãČL'aČT'aČC'aČrāR'ařČÜaČl'ãČrāC'd'aČşaaĀđaČT'aČijaaČaāAđeřaňořžaR'eřč;aĀC

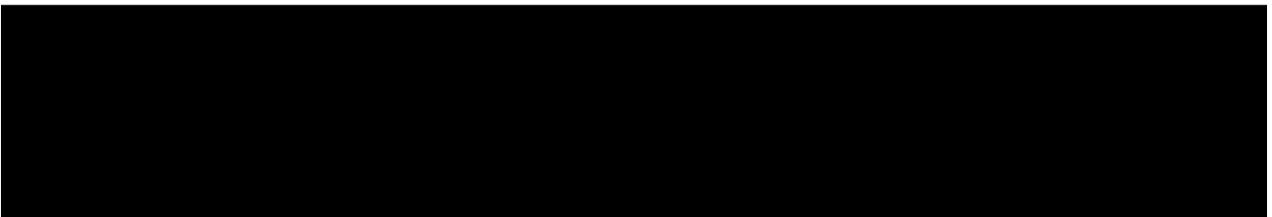
ãČiaČIjaaČC'aČL'aAđaČn'aČijaaČL'aČl'aČşaaČrāČšaeňG'ařžaĀčN'aAšaaČAaĀAňaĀažeāyN'aAđaČL'aAđeřAařsrdfāČš/  
robot\_description\_semanticsAňeřaňořžaĀčN'ařEeřAaĀňaAčaČL'ač'aAžaĀC



ãAŞaAősrdfāAřmoveitAňgaaČÜaČl'ãČşaaČrāČS'eřN'aAřeřAšaaČAaAňaĀřeřČEňašaAňaAđaAžaĀč'aAč'aAžaĀňaĀřf  
ãAŞaAőaČL'aAřeřAřsrdfāČS'eřG'ařN'TcZD'aAňeřaňořžaĀčN'aAšaaČAaAđaČz'aČrāČl'aČL'aAň  
package://jsk\_interactive\_marker/scripts/semantic\_robot\_state\_generator.py aAļaAđaAęeřR'R'ařžaAřaČN'aAęaAđaAč'aAžaĀC

ãAŞaAőaČz'aČf'aČL'aAđeřaňořžaAřaČN'ařsrdfāAřworld\_joinたAļaAđaAřeřořeřČşaaČyaaČgaaČ'd'aČşaaČL'aAňaĀčn'aČijaaČ  
ãAŞaĀčN'aAřmoveitelžařUaAđašiařR'ařL'čařAřiřyAeřG'ařaAřaĀčN'aAšaaČAaĀAřmoveit\_configAřgčTšaeřL'R'ařAđaAřsrdfāČšse  
pyaňořeřčařČrāČL'aAňcTílaAđaČN'aAšaaĀčČaAđgaaAřaňořžaĀč'aAžaĀC

euslispāAňaČL'moveit\_msgs/DisplayRobotStateãČS'čTšaeřL'R'ařaĀčN'eřUćařTřaAřaAđaAřangle-vector-to-disp  
robot-state.l'aAňaAęeřR'R'ařžaAřaČN'aAęaAđaAč'aAžaĀC aAšaaČN'aČšcTílaAđaČN'aAňaAđaAňaĀažeāyN'aAđaČL'aAđeřAaňaAđaAřeřviziřyL'a



### 3.1.1 æšíaeřRćCz

rvizāAřRobotStateãČS'eř; aL'aAžaĀčN'aL'ařiřrobot\_description\_stateãAňeřaňořžaAřaČN'ařeřAđaČN'ařfEeřAaňaAňaA  
aAňaAřeřAňaAđaAřaĀčn'aČijaaČL'aČl'aČşaaČrāAđeřařč'd'žařAňaAřaĀňaAřaĀřaAřaČL'ač'aAžaĀC

## **3.2 camera\_info\_publisher**

`camera_info_publisher` provides camera info topics to an image or pointcloud without camera info.

### **3.2.1 Parameters**

- `~yaml_filename` (String, default: ``)  
Path to yaml file which has camera info information.
- `~frame_id` (String, default: `camera`)  
Frame id of camera info.
- `~parent_frame_id_` (Bool, default: `base_link`)  
Frame id of interactive marker.
- `~sync_pointcloud` (String, default: `false`)  
Synchronize camera info to pointcloud. If both `~sync_pointcloud` and `~sync_image` are not specified, camera info is published at a static rate.
- `~sync_image` (String, default: `false`)  
Synchronize camera info to image.
- `~static_rate` (Double, default: `30.0`)  
Static rate at which camera info is published. If both `~sync_pointcloud` and `~sync_image` are not specified, camera info is published at a static rate.
- `~width` (Double, default: `640`)  
Width of published camera info. This parameter is enabled when `~yaml_filename` is not specified. This parameter can be changed by dynamic reconfigure.
- `~height` (Double, default: `480`)  
Height of published camera info. This parameter is enabled when `~yaml_filename` is not specified. This parameter can be changed by dynamic reconfigure.
- `~f` (Double, default: `525`)  
F of published camera\_info. This parameter is enabled when `~yaml_filename` is not specified. This parameter can be changed by dynamic reconfigure.

### **3.2.2 Subscribing Topics**

- `~input` (sensor\_msgs/Image or sensor\_msgs/Pointcloud2)  
Image or pointcloud whose camera info is published.

### 3.2.3 Publishing Topics

- `~camera_info` (`sensor_msgs/CameraInfo`)  
Camera info which has the same timestamp as the input topic.

### 3.2.4 Sample



## 3.3 marker\_6dof

`marker_6dof` provides interactive marker to control a marker of primitive shape or mesh shape.

### 3.3.1 Parameters

- `~object_type` (`String, default: sphere`)  
Type of object shape. cube, sphere, line, and mesh are available.
- `~frame_id` (`String, default: /map`)  
Frame id of marker.
- `~publish_tf` (`Bool, default: False`)  
Tf of marker pose is published if true.

- `~tf_frame` (String, default: `object`)  
frame id of published tf. This value is used only when `~publish_tf` is true.
- `~tf_duration` (Double, default: 0.1)  
Time interval of published tf. This value is used only when `~publish_tf` is true.
- `~publish_pose_periodically` (Bool, default: False)  
Pose of marker is published periodically if true. Pose topic is published only when marker is moved via Rviz if false.
- `~object_x` (Double, default: 1.0)
- `~object_y` (Double, default: 1.0)
- `~object_z` (Double, default: 1.0)  
X, Y, Z scale of object.
- `~object_r` (Double, default: 1.0)
- `~object_g` (Double, default: 1.0)
- `~object_b` (Double, default: 1.0)
- `~object_a` (Double, default: 1.0)  
Red, Green, Blue and Alpha value of object.
- `~initial_x` (Double, default: 0.0)
- `~initial_y` (Double, default: 0.0)
- `~initial_z` (Double, default: 0.0)  
Initial X, Y, Z position of marker.
- `~initial_orientation` (Vector of Double, default: [0.0, 0.0, 0.0, 1.0])  
Initial orientation of marker described in quaternion.

### 3.3.2 Subscribing Topics

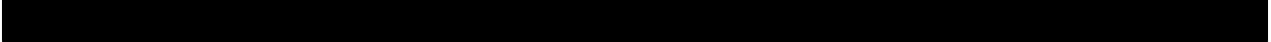
- `~feedback` (visualization\_msgs/InteractiveMarkerFeedback)
- `~move_marker` (geometry\_msgs/PoseStamped)

You can control markers through topics above.

### 3.3.3 Publishing Topics

- `~update` (visualization\_msgs/InteractiveMarkerUpdate)
- `~update_full` (visualization\_msgs/InteractiveMarkerInit)  
Current marker state
- `~pose` (geometry\_msgs/PoseStamped)  
Pose of marker. You can select publishing policy via `~publish_pose_periodically`.
- `/tf` (tf2\_msgs/TFMessage)  
Tf of marker pose. Available only when `~publish_tf` is true.

### 3.3.4 Sample



## 3.4 polygon\_marker

`polygon_marker` is a simple code to provide interactive marker to select one polygon out of multiple polygons represented in `jsk_recognition_msgs/PolygonArray`.

### 3.4.1 Subscribing Topics

- `~polygon_array (jsk_recognition_msgs/PolygonArray)`  
Input polygons

### **3.4.2 Publishing Topics**

- `~selected_index (jsk_recognition_msgs/Int32Stamped)`  
Selected index of the polygon.
- `~selected_polygon (geometry_msgs/PolygonStamped)`  
Selected polygon as geometry\_msgs/PolygonStamped
- `~selected_polygon_array (jsk_recognition_msgs/PolygonArray)`  
Selected polygon as jsk\_recognition\_msgs/PolygonArray.

## **3.5 transformable\_markers\_client.py**

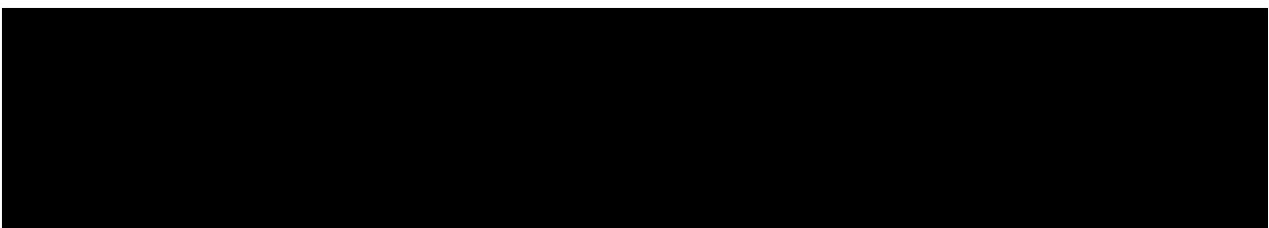
`transformable_markers_client.py` has features below:

- Insert markers to `transformable_server_sample`
- Auto save of user interaction of the markers
- Publish topics of reusable msgs after conversion from the markers (ex. BOX -> BoundingBox)

### **3.5.1 Parameters**

- `~config_file (String, required)`

Config file to insert markers, and auto save the interaction. The format is like below:



- `~config_auto_save (Bool, default: True)`

Enable the feature to save the config automatically.

### 3.5.2 Required ROS name

- `~server`  
Node name of `transformable_server_sample` server.

### 3.5.3 Publishing Topics

- `~output/boxes (jsk_recognition_msgs/BoundingBoxArray)`  
Converted boxes from marker: BOX -> BoundingBox.

### 3.5.4 Sample



## 3.6 `transformable_server_sample`

`transformable` provides interactive marker to control some object models.

### 3.6.1 Parameters

- `~server_name`  
Name of interactive server.
- `~use_parent_and_child (default: false)`  
Flag for using ParentAndChildInteractiveServer.  
If true, you can use associate markers like below:

If parent\_topic\_name==empty, it uses self server, and only 1 hierarchy is supported.

- ~display\_interactive\_manipulator (Bool, default: true)  
Flat to show the 6dof interactive manipulator for all objects.
- ~display\_interactive\_manipulator\_only\_selected (Bool, default: false)  
Flag to show the 6dof interactive manipulator only for the selected object. This flag does nothing if ~display\_interactive\_manipulator is false.
- ~display\_description\_only\_selected (Bool, default: false)  
Flag to show the description only for the selected object.

### 3.6.2 Usage

then, in different terminal You can insert box marker by this command

You can insert model by this command

### 3.6.3 Topics

You can control markers through topics below (Please Read msg Structure by rosmsg show or some other) (The default topic name is /simple\_marker, server name is the same)

- /simple\_marker/set\_color
- /simple\_marker/feedback [visualization\_msgs/InteractiveMarkerFeedback]
- /simple\_marker/set\_radius
- /simple\_marker/set\_control\_relative\_pose
- /simple\_marker/add\_pose\_relative [geometry\_msgs::Pose]
- /simple\_marker/set\_z
- /simple\_marker/set\_x
- /simple\_marker/set\_y
- /simple\_marker/add\_pose [geometry\_msgs::Pose]
- /simple\_marker/set\_control\_pose [geometry\_msgs::PoseStamped]

- /simple\_marker/set\_pose [geometry\_msgs::PoseStamped]

You can get marker info by topics below

- /simple\_marker/marker\_dimensions [jsk\_interactive\_marker/MarkerDimensions]
- /simple\_marker/pose [geometry\_msgs/PoseStamped]
- /simple\_marker/focus\_object\_marker\_name [std\_msgs/String]
- /simple\_marker/pose\_with\_name [jsk\_interactive\_marker/PoseStampedWithName]
- /simple\_marker/focus\_marker\_name\_text [jsk\_rviz\_plugins/OverlayText]
- /tf [tf2\_msgs/TFMessage] (with marker name, tf is published)
- /simple\_marker/focus\_marker\_pose\_text [jsk\_rviz\_plugins/OverlayText]

### 3.6.4 Services

You can control markers through topics below

- /simple\_marker/request\_marker\_operate -> for inserting marker
- /simple\_marker/set\_focus
- /simple\_marker/set\_color
- /simple\_marker/set\_control\_pose
- /simple\_marker/set\_parameters
- /simple\_marker/set\_pose
- /simple\_marker/set\_dimensions
- /simple\_marker/set\_parent\_marker
- /simple\_marker/hide

You can get marker info through topics below

- /simple\_marker/get\_control\_pose
- /simple\_marker/get\_color
- /simple\_marker/get\_focus
- /simple\_marker/get\_dimensions
- /simple\_marker/get\_type
- /simple\_marker/get\_pose
- /simple\_marker/get\_existence

## **3.7 urdf\_model\_marker**

`urdf_model_marker` provides interactive marker to control robot model.

### **3.7.1 Parameters**

- `~server_name`  
Name of nteractive server.
- `~use_dynamic_tf` (default: `true`)  
Use `dynamic_tf_publisher` if it is true.
- `~model_config`  
Collection of parameters.
  - `name`
  - `description`
  - `scale`
  - `pose`
  - `offset`

- use\_visible\_color
- frame\_id
- registration
- fixed\_link
- model
- use\_robot\_description
- model\_param
- robot
- mode
- initial\_joint\_state
  - \* name
  - \* position
- display

### **3.7.2 Sample**



# Chapter 4

## Tips about visualization

### 4.1 Record rviz

#### 4.1.1 kazam

You can use kazam to record desktop movie easily.

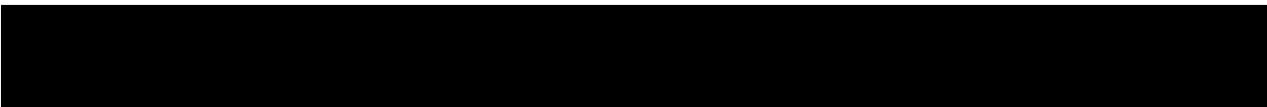


#### 4.1.2 glc

You can use glc to record OpenGL rendering. glc is the best way to record OpenGL application because it can record movie efficiently. glc record OpenGL rendering to a special file called .glc and you can convert the .glc into several movie format. It is also good for irtviewer.

#### Install

You can install glc via ppa package. (see this tutorial)



#### Capture

You can use glc as wrap command like:



If you want to use it in launch file, use `launch-prefix` attribute:



You need to type Shift + F8 to start and stop capturing. (see this tutorial for detail)

### **Convert to movie**

You can use `glc_encode.sh` under `jsk_tools`.