CS 309: Autonomous Intelligent Robotics
FRI I

Lecture 14:
OpenCV
Rviz

http://justinhart.net/teaching/2019_spring_cs309/

Basic computer vision ideas in OpenCV

- The basics
  - Color channels
  - Color channel subtraction
  - Thresholding
  - Contour Detection
  - Masking
- These are some of the most basic tools in computer vision, but will enable you to do some simple object detection and tracking.

OpenCV and ROS use different formats

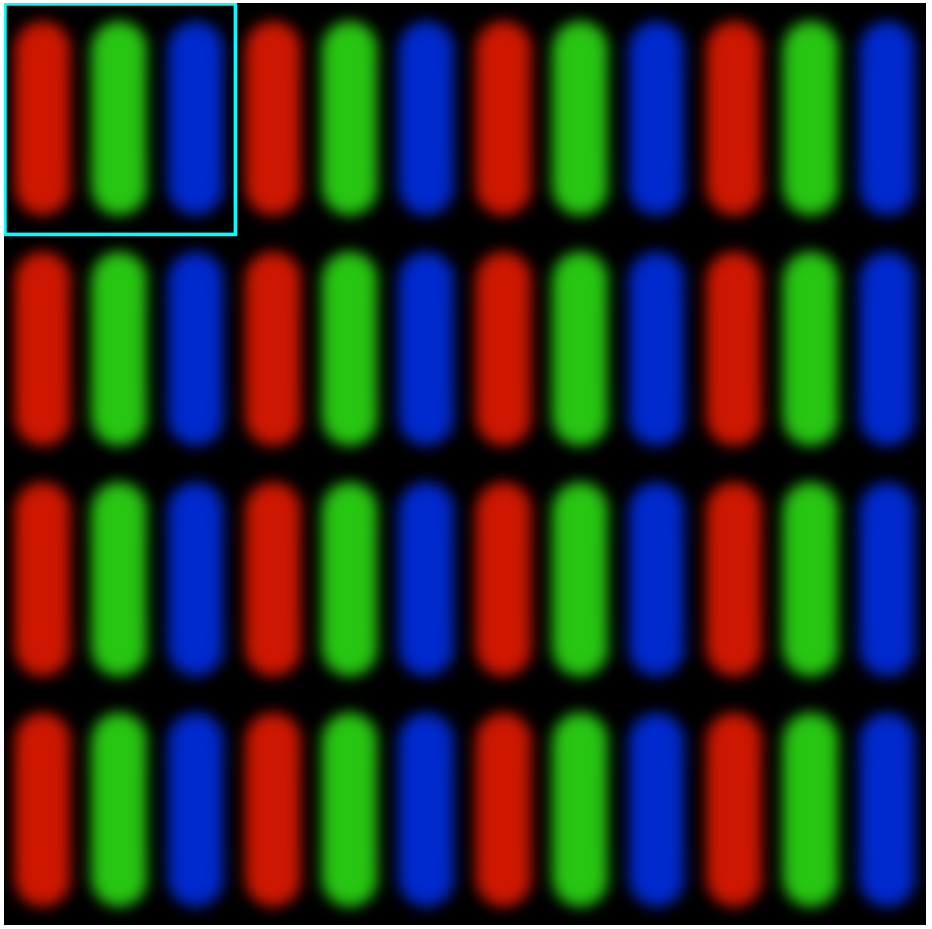
- cv_bridge helps solve this

Color Channels

- Color images can be represented under several different systems.
  - BGR → Blue, Green, Red
  - HSV → Hue, Saturation, Value
  - Others get a bit more complex
  - Today, we focus on BGR

BGR

- In BGR, each pixel gets a color intensity for each channel
- The blend of the colors blue, green, and red becomes the final color represented



OpenCV and BGR

- In OpenCV, images are stored in a matrix type
  - cv::Mat
- A matrix has rows and columns
  - For an image, this is how tall and how wide the image is
- In OpenCV, each cell of the matrix can have more than one channel, and the matrix takes on a type that represents this
- BGR images are stored in CV_8UC3
  - OpenCV, 8 bits per channel, unsigned character, 3 channels

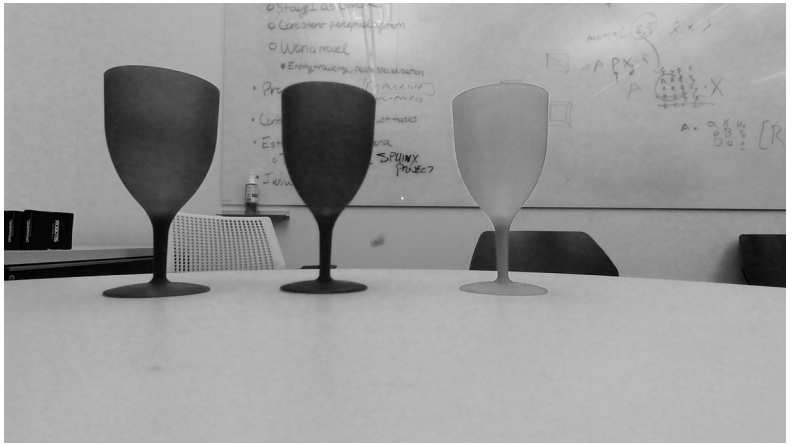
Color Values

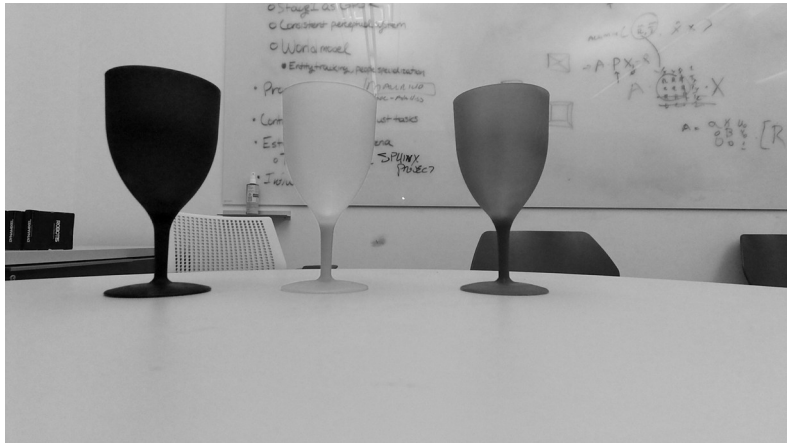
- An unsigned character is 8 bits long
  - 0..255
  - So the highest intensity is 255, the lowest is 0
  - As the intensity gets higher, the color in that channel gets brighter

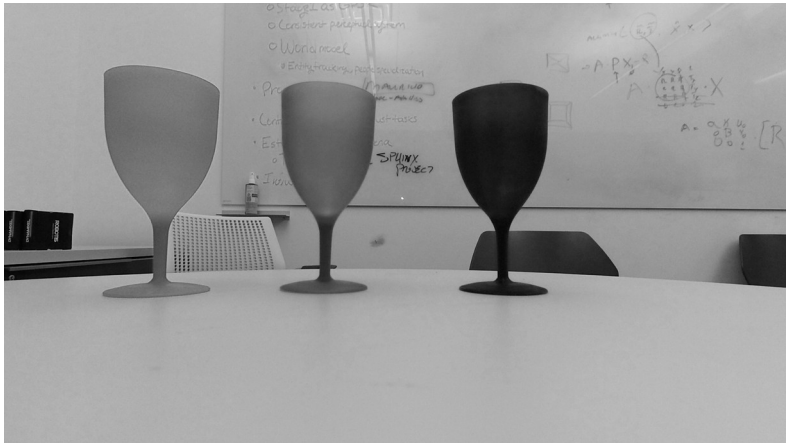
- cv::split()
  - Allows us to break an image with several channels into several 1 channel images

```
std::vector<cv::Mat> chans;
split(image, chans);
```

Input image, as 3 channels







Color Channel Images

- As the intensity goes up, the channel's greyscale image becomes brighter
- We can use this for a technique called color blob detection
- In this example, we will find the blue cup by finding the bluest pixels

Color Channel Subtraction

- cv::subtract()
  - Allows you to subtract one cv::Mat from another
  - cv::Mat bMinusG;
  - cv::subtract(chans[0], chans[1], bMinusG);

Color Subtracted Images

- Blue channel minus red channel

- Blue channel minus green channel

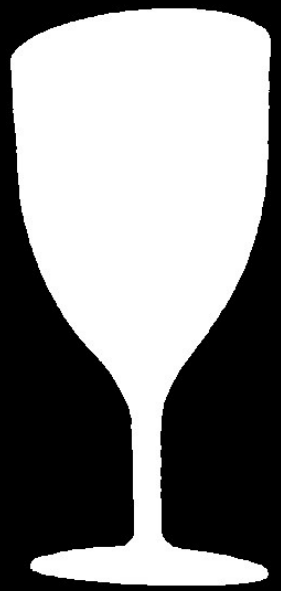
Picking out the blue pixels

- We see that Blue minus Red gives us really bright pixels where the blue cup is, so we'll simply focus on that

Image Thresholding

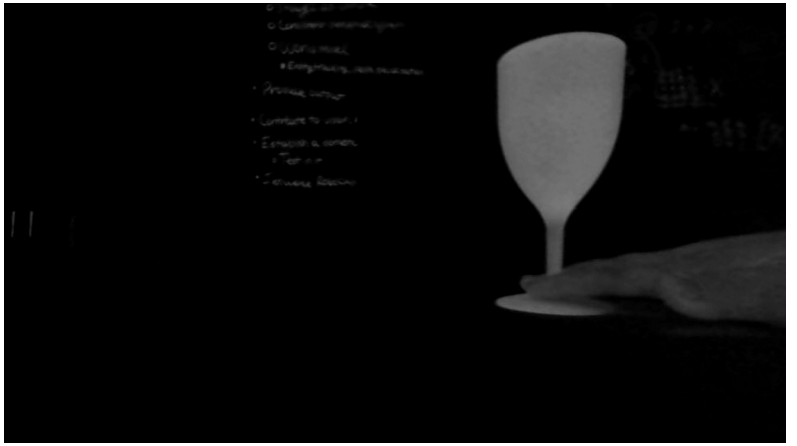
- There are other illuminated pixels in the image, but the brightest ones are now the cup.
  - So we will pick the pixels that are only at least as bright as some value
- This is image thresholding
  - You can specify both a minimum and a maximum threshold
  - cv::threshold(input_image, output_image, threshold_value, value_when_above_threshold, threshold_type)
  - For now, we will use only cv::THRESH_BINARY
    - It is or is not above the threshold

```
cv::Mat bThresh;
cv::threshold(bMinusR, bThresh, 50, 255, cv::THRESH_BINARY);
```



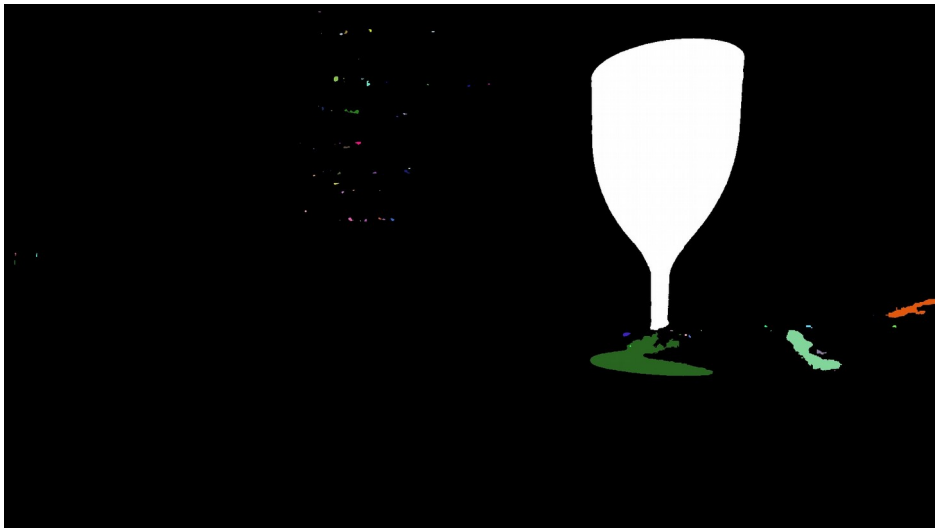
Contour Detection

- Looks for closed image contours in a scene
  - These are the “blobs” in the image, connected areas in the threshold image.
  - This is more obvious in the next set of images



Area in a contour

- cv::contourArea(contours[i])
  - Looks for the size of a contour
  - In your program, look for the biggest contour and track it, to get rid of noise
  - In this example, it looks like this



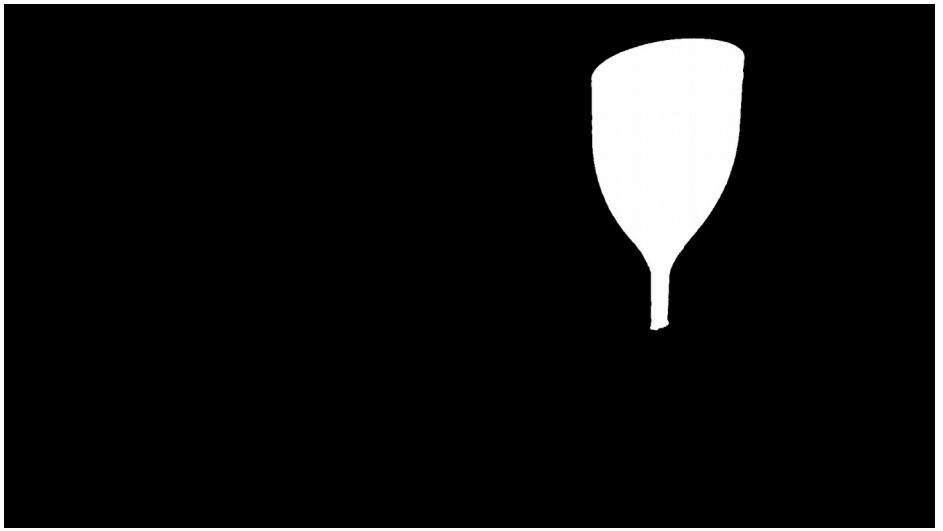


Image Masking

- Masking is using only certain pixels
- A mask is computed as a 1-channel image
- In the example, this happens here

```
cv::Mat mask = cv::Mat::zeros(image.rows, image.cols, CV_8UC1);
drawContours( mask, contours, maxSizeContour, cv::Scalar(255),
cv::LineTypes::FILLED, 8, hierarchy );
```

- copyTo can be used with a mask like this image.copyTo(blueCupImg, mask);

For your homework..

- You will create your own ROS package which puts the blue, green, and red cups together
  - This package must build under catkin_make or catkin build with g++
- The three separate cups get published on ROS topics
- The three cups together get published on a ROS topic in a composite image
- This will stream using data from three_cups.bag
- Publishing the image topics will use cv_bridge

ROS Workspace Creation Review

- mkdir <ros_workspace_name>
- cd <ros_workspace_name>
- mkdir src
- cd src
- catkin_init_workspace

ROS Package Creation Review

- We went over this before, so we'll only hit the high points

catkin_create_pkg

- Creates a package template that you can fill in
  - catkin_create_pkg <package_name> roscpp rospy std_msgs <other dependencies if you need them>
- In your homework, you will use
  - catkin_create_pkg hw3 roscpp rospy std_msgs sensor_msgs cv_bridge image_transport
  - Consider this a free tip on how to solve your homework
- Should be run from your workspace’s src directory

CMakeLists.txt

- Used to build your software
  - We can pick through this file if needed
- The version created by catkin_create_pkg is only a template, you will need to uncomment and modify the lines that you need
  - # add_executable(${PROJECT_NAME}_node src/hw3_node.cpp)
  - # target_link_libraries(${PROJECT_NAME}_node
${catkin_LIBRARIES}
)
  - Possibly others

package.xml

- The version made by catkin_create_pkg is probably actually correct. Your implementation may vary
- package.xml is your manifest file
  - It tells ROS how to treat your package
    - Name
    - License
    - Maintainer
    - If it requires other packages in order to build or run it

“catkin_make” and “catkin build”

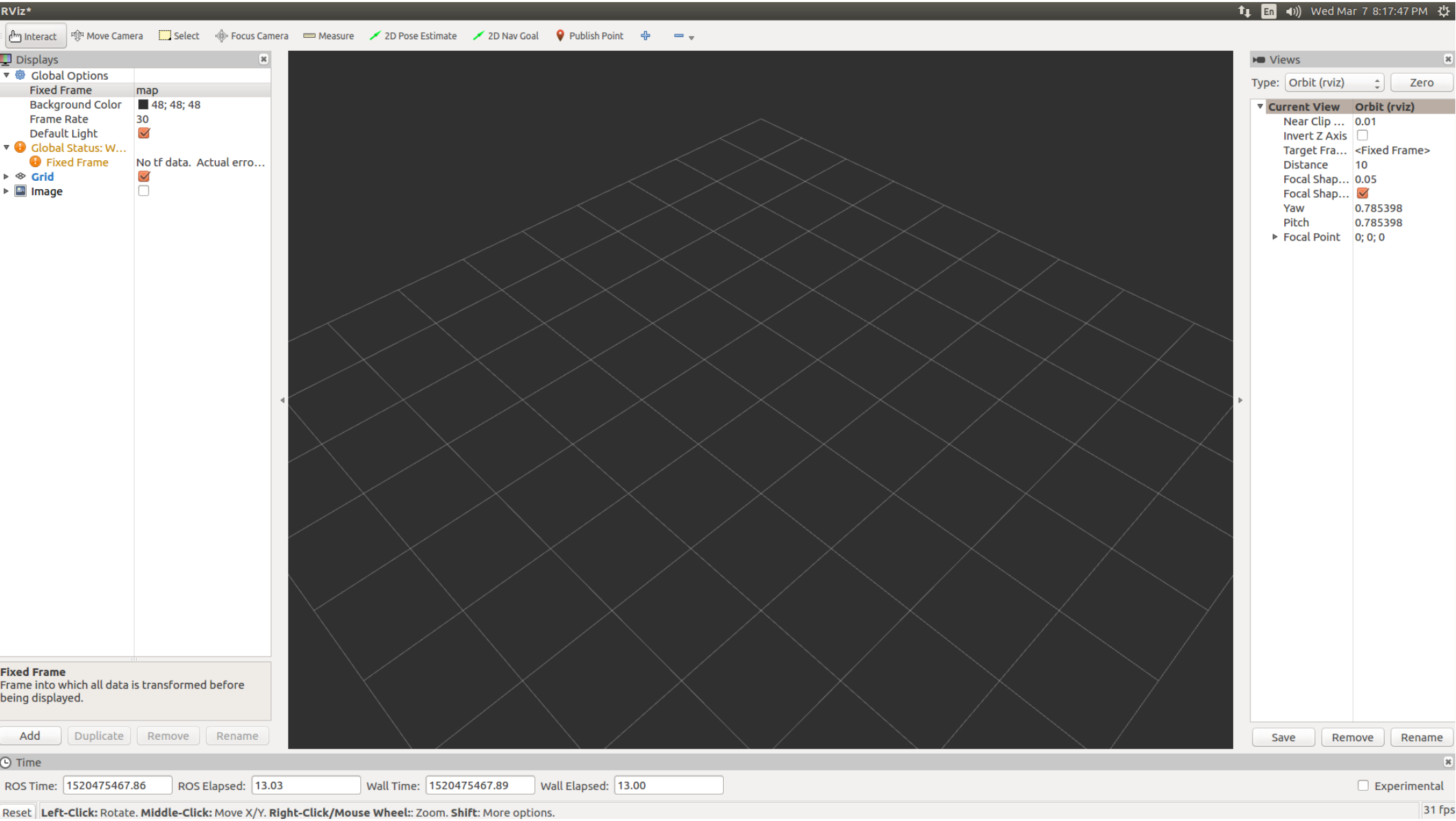
- Either is run from the top of your workspace
- Will build your software into your ROS workspace
- You will need to source devel/setup.bash to include your workspace into your ROS environment
- Once you have done that, you should be able to run your homework from rosrun hw3 <program_name>
- Remember to run roscore before any program that is not in a launch file, including rviz or rosbag

rosbag

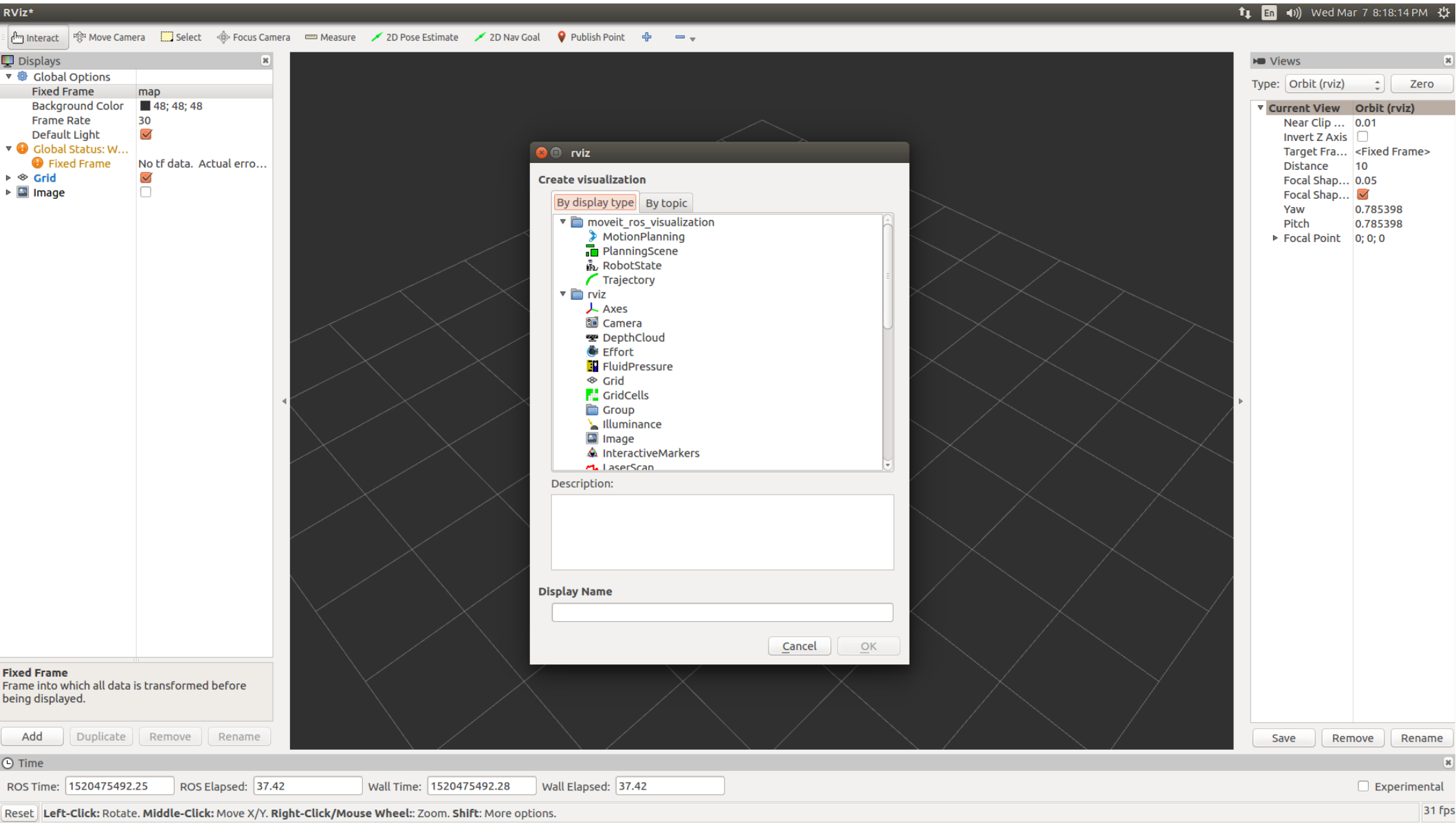
- You will use three_cups.bag for your homework
- rosbag records of plays back pre-recorded data from ROS topics
- rosbag play -l three_cups.bag
  - -l makes it run the bag in a loop

rviz – The ROS Visualizer

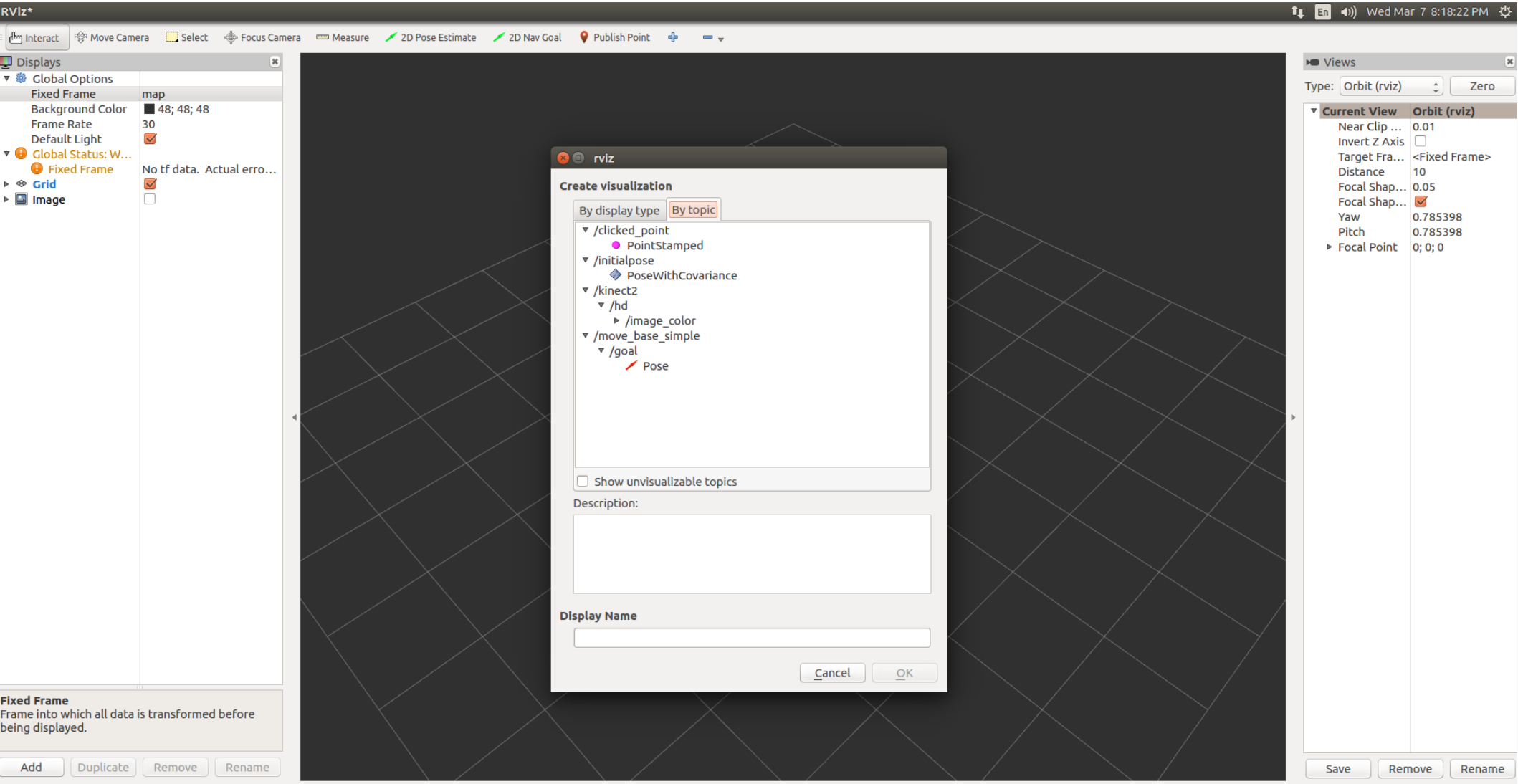
- rosrun rviz rviz
- What you get should look something like this.


RViz*
En
Wed Mar 7 8:17:47 PM
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame
map
Background Color
48; 48; 48
Frame Rate
30
Default Light
Global Status: W...
Fixed Frame
No tf data. Actual erro...
Grid
Image
Fixed Frame
Frame into which all data is transformed before being displayed.
Add
Duplicate
Remove
Rename
Views
Type:
Orbit (rviz)
Zero
Current View
Orbit (rviz)
Near Clip ...
0.01
Invert Z Axis
Target Fra...
<Fixed Frame>
Distance
10
Focal Shap...
0.05
Focal Shap...
Yaw
0.785398
Pitch
0.785398
Focal Point
0; 0; 0
Save
Remove
Rename
Time
ROS Time:
1520475467.86
ROS Elapsed:
13.03
Wall Time:
1520475467.89
Wall Elapsed:
13.00
Experimental
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
31 fps

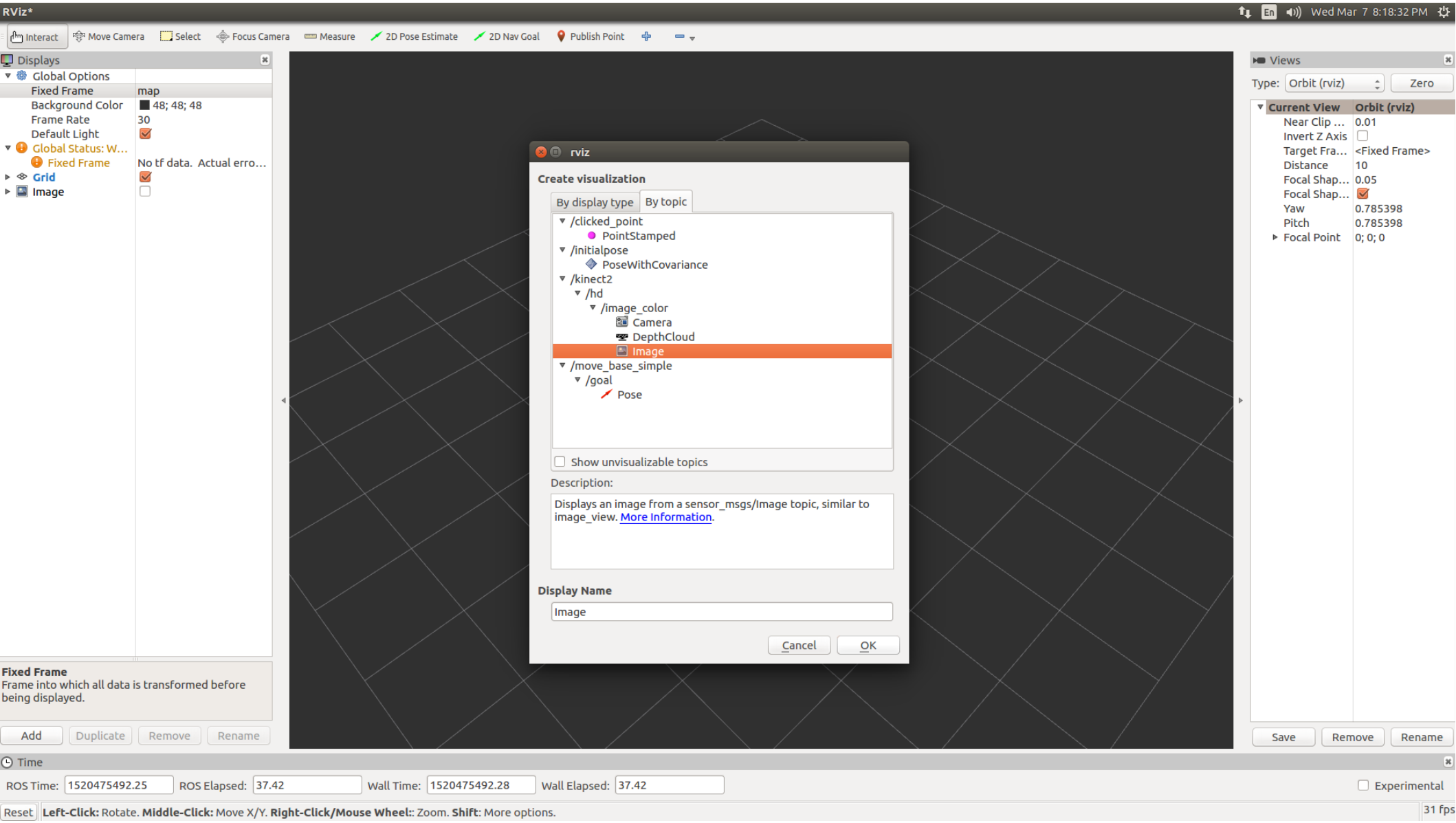
- Clicking the add button will allow you to add things to visualize.


RViz*
Wed Mar 7 8:18:14 PM
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame
map
Background Color
48; 48; 48
Frame Rate
30
Default Light
Global Status: W...
Fixed Frame
No tf data. Actual erro...
Grid
Image
Fixed Frame
Frame into which all data is transformed before being displayed.
Add
Duplicate
Remove
Rename
rviz
Create visualization
By display type
By topic
moveit_ros_visualization
MotionPlanning
PlanningScene
RobotState
Trajectory
rviz
Axes
Camera
DepthCloud
Effort
FluidPressure
Grid
GridCells
Group
Illuminance
Image
InteractiveMarkers
LaserScan
Description:
Display Name
Cancel
OK
Views
Type:
Orbit (rviz)
Zero
Current View
Orbit (rviz)
Near Clip ...
0.01
Invert Z Axis
Target Fra...
<Fixed Frame>
Distance
10
Focal Shap...
0.05
Focal Shap...
Yaw
0.785398
Pitch
0.785398
Focal Point
0; 0; 0
Save
Remove
Rename
Time
ROS Time:
1520475492.25
ROS Elapsed:
37.42
Wall Time:
1520475492.28
Wall Elapsed:
37.42
Experimental
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
31 fps

- “By topic” will list the available topics.


RViz*
Wed Mar 7 8:18:22 PM
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame
map
Background Color
48; 48; 48
Frame Rate
30
Default Light
Global Status: W...
Fixed Frame
No tf data. Actual erro...
Grid
Image
Fixed Frame
Frame into which all data is transformed before being displayed.
Add
Duplicate
Remove
Rename
rviz
Create visualization
By display type
By topic
/clicked_point
PointStamped
/initialpose
PoseWithCovariance
/kinect2
/hd
/image_color
/move_base_simple
/goal
Pose
Show unvisualizable topics
Description:
Display Name
Cancel
OK
Views
Type:
Orbit (rviz)
Zero
Current View
Orbit (rviz)
Near Clip ...
0.01
Invert Z Axis
Target Fra...
<Fixed Frame>
Distance
10
Focal Shap...
0.05
Focal Shap...
Yaw
0.785398
Pitch
0.785398
Focal Point
0; 0; 0
Save
Remove
Rename
Time
ROS Time:
1520475492.25
ROS Elapsed:
37.42
Wall Time:
1520475492.28
Wall Elapsed:
37.42
Experimental
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
31 fps

- Expand until you see what you are interested in.


RViz*
Wed Mar 7 8:18:32 PM
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame
map
Background Color
48; 48; 48
Frame Rate
30
Default Light
Global Status: W...
Fixed Frame
No tf data. Actual erro...
Grid
Image
Fixed Frame
Frame into which all data is transformed before being displayed.
Add
Duplicate
Remove
Rename
rviz
Create visualization
By display type
By topic
/clicked_point
PointStamped
/initialpose
PoseWithCovariance
/kinect2
/hd
/image_color
Camera
DepthCloud
Image
/move_base_simple
/goal
Pose
Show unvisualizable topics
Description:
Displays an image from a sensor_msgs/Image topic, similar to image_view. More Information.
Display Name
Image
Cancel
OK
Views
Type:
Orbit (rviz)
Zero
Current View
Orbit (rviz)
Near Clip ...
0.01
Invert Z Axis
Target Fra...
<Fixed Frame>
Distance
10
Focal Shap...
0.05
Focal Shap...
Yaw
0.785398
Pitch
0.785398
Focal Point
0; 0; 0
Save
Remove
Rename
Time
ROS Time:
1520475492.25
ROS Elapsed:
37.42
Wall Time:
1520475492.28
Wall Elapsed:
37.42
Experimental
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
31 fps

- Clicking “Okay” should add it to the interface.

RViz*

Wed Mar 7 8:18:42 PM

Interact | Move Camera | Select | Focus Camera | Measure | 2D Pose Estimate | 2D Nav Goal | Publish Point

Displays

Global Options	
Fixed Frame	map
Background Color	48; 48; 48
Frame Rate	30
Default Light	☑
Global Status: W…	
Fixed Frame	No tf data. Actual erro…
Grid	☑
Image	☐
Image	☑

Fixed Frame
Frame into which all data is transformed before being displayed.

Add | Duplicate | Remove | Rename

Image

Views

Type: Orbit (rviz) | Zero

Current View	Orbit (rviz)
Near Clip …	0.01
Invert Z Axis	☐
Target Fra…	<Fixed Frame>
Distance	10
Focal Shap…	0.05
Focal Shap…	☑
Yaw	0.785398
Pitch	0.785398
Focal Point	0; 0; 0

Save | Remove | Rename

Time

ROS Time: 1520475522.11 | ROS Elapsed: 67.28 | Wall Time: 1520475522.15 | Wall Elapsed: 67.18 | ☐ Experimental

Reset | 31 fps

- You can rearrange and resize windows as appropriate.

RViz*

Wed Mar 7 8:18:53 PM

Interact | Move Camera | Select | Focus Camera | Measure | 2D Pose Estimate | 2D Nav Goal | Publish Point

Displays

Global Options	
Fixed Frame	map
Background Color	48; 48; 48
Frame Rate	30
Default Light	☑
Global Status: W…	
Fixed Frame	No tf data. Actual erro…
Grid	☑
Image	☐
Image	☑

Fixed Frame
Frame into which all data is transformed before being displayed.

Add | Duplicate | Remove | Rename

Image

Views

Type: Orbit (rviz) | Zero

Current View	Orbit (rviz)
Near Clip …	0.01
Invert Z Axis	☐
Target Fra…	<Fixed Frame>
Distance	10
Focal Shap…	0.05
Focal Shap…	☑
Yaw	0.785398
Pitch	0.785398
Focal Point	0; 0; 0

Save | Remove | Rename

Time

ROS Time: 1520475533.99 | ROS Elapsed: 79.16 | Wall Time: 1520475534.03 | Wall Elapsed: 79.13 | ☐ Experimental

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

- Use rviz to help develop and debug your homework.

What should my program do?

- You should write 1 (and only 1) ROS node
- It should publish 4 topics
  - /color_filter/blue_cup
  - /color_filter/green_cup
  - /color_filter/red_cup
  - /color_filter/cups
- The first three should show only the blue, green, and red cup, respectively.
- The third should show all three together.

How should my program do this?

- Finding the BLUE cup is demonstrated in the example on justinhart.net
  - You may need to modify your package.xml and CMakelists.txt as per the Piazza discussion
- Finding the RED and GREEN cups is a variation on this

How should my program do this?

- /color_filter/cups contains all three, though!
- RIGHT! I'm not going to tell you *exactly* how to do this, because it would make the homework too easy for you to learn anything.
- But you will use cv::bitwise_or to do it
  - And if you Google cv::bitwise_or, and understand how you found the blue,green, and red cups; the example for bitwise_or is almost exact directions on how this works.

So.. display the cups, right?

- NO!!
  - Publish a ROS TOPIC for each of the the blue, green, and red cups, respectively, and one containing all three.
  - You should be able to see this topic using rviz
  - In fact, turn off the cv::imshows in your c++ code before you submit (unless you've already submitted).

How do I publish the ROS topic?

- cv_bridge
  - http://wiki.ros.org/cv_bridge/Tutorials/UsingCvBridgeToConvertBetweenROSImagesAndOpenCVImages
  - Then publish the topic as in our previous lectures.
  - See also: https://stackoverflow.com/questions/27080085/how-to-convert-a-cvmat-into-a-sensor-msgs-in-ros

My ROS topic complains that I’ve advertised more than once

- Call advertise() in your main, and publish() in your callback.
  - publish() sends the image
  - advertise() says that you will publish on a topic

My ROS topic complains that I’ve advertised more than once

- Call advertise() in your main, and publish() in your callback.
  - publish() sends the image
  - advertise() says that you will publish on a topic
    - And you can’t advertise the same topic more than once per node.