

ABOUT ME:

Hi, I'm Sree Balaji S

Roll no: EC23B1041

Well, I'm currently a 2nd year ECE student and doing Data science on the side, I'll be completing diploma in programming soon.

I like studying science and given the time I'll study and other fields of science too.

Coding skills are essential requirement in any field in this modern age.

Coding for me is implementing my ideas in a logical and instructional manner so the hardware can understand.

What drew me into AUV?

During my end of first year, I was quite negligent and was not much aware of the technical clubs. During the onset of my second year, I began talking about auv with some people on a regular basis. During many such conversations my interest in joining Auv drew significantly. In my opinion Auv is the most well-disciplined club in all of college. I'm very motivated to put in the heavy hours and contributing significantly to the club. Its auv or no other clubs for me.

Here I will summarize Ros and finite state machines. In the questions folders, Readme.md I'll provide info on how I tackled the problem statement. I will try to keep it short. Also, to mention is that my code is not written to be robust and modular. All these codes were written while learning all the new stuff, so I couldn't focus on modularity and robustness.

ROS:

Layer on top OS we can say although not accurate. Auv uses ros mainly as a comms interface.



Data Distribution Service

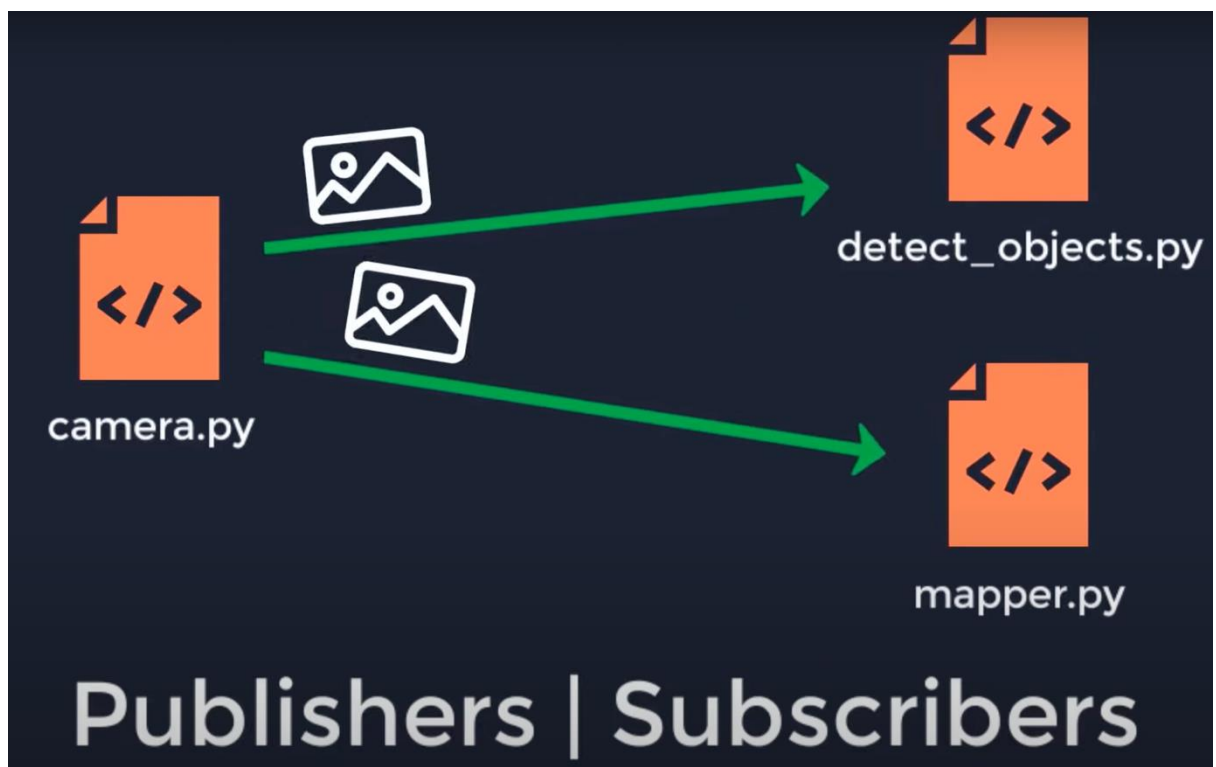
Communication Pipeline Interface

The pipes are given a name and referred to as topic.

Scripts can be called as nodes.

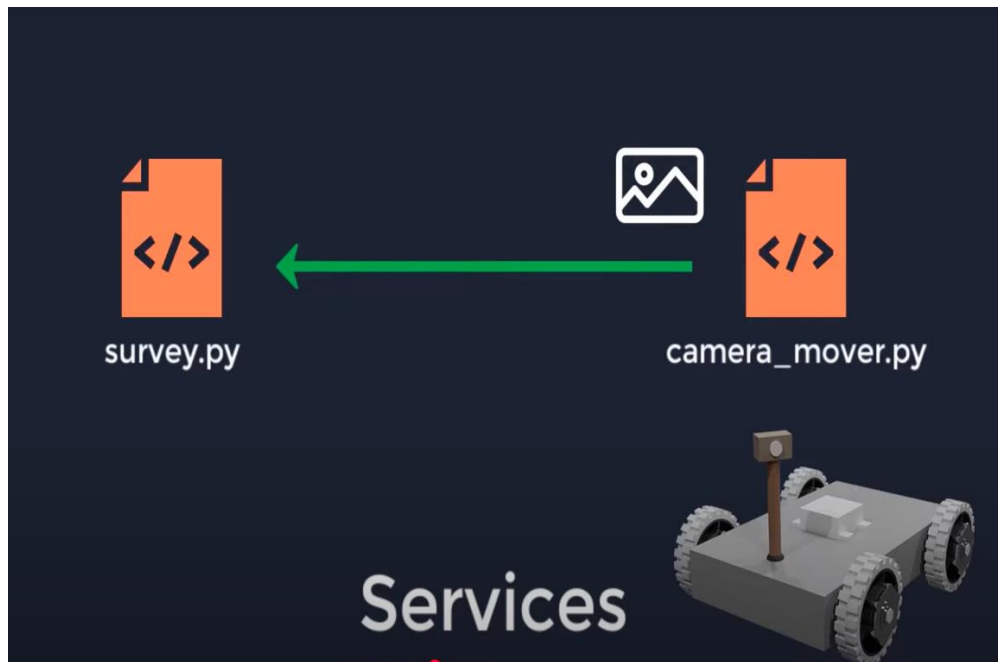
There are 3 main types of DDS:

1) Publisher and Subscriber:



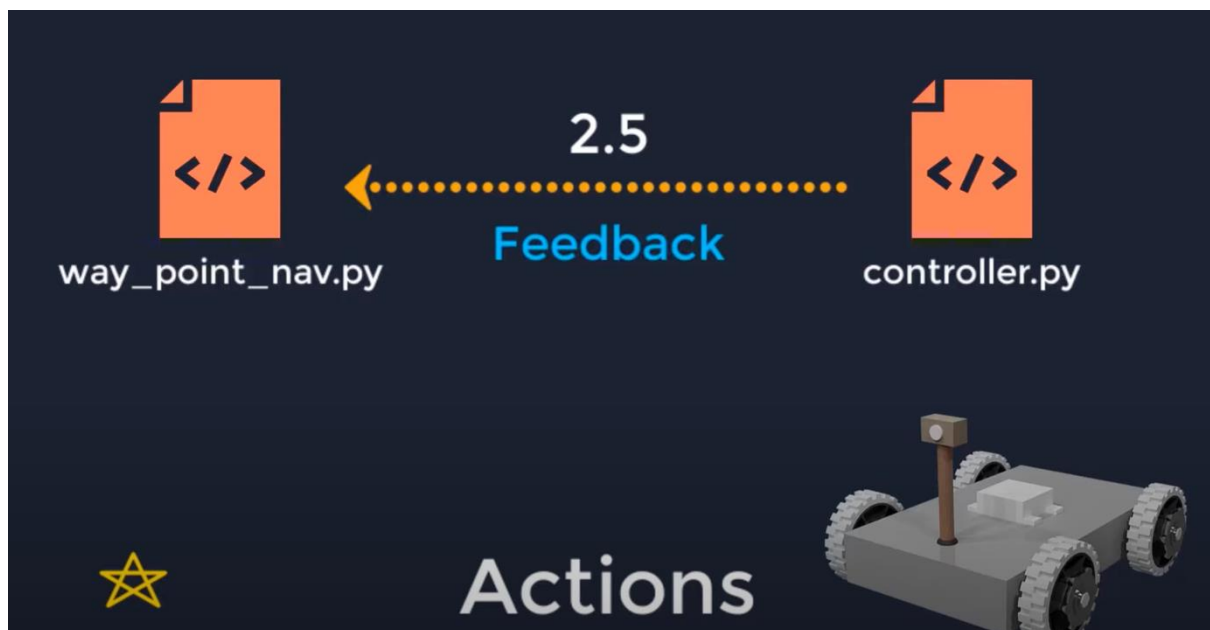
Similar to the YouTube model.

2) Services:



After completing a survey, a response is sent to the requested node.

3)Actions:



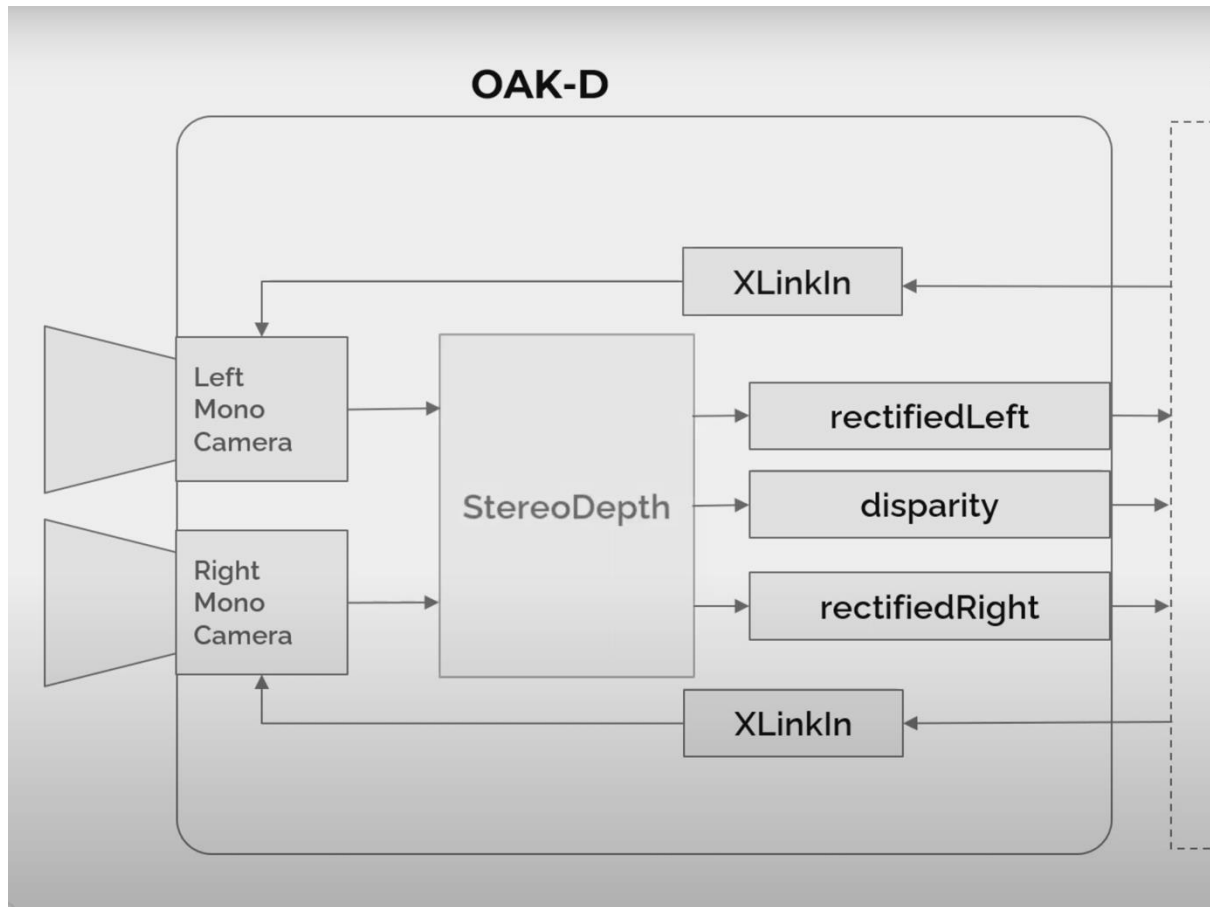
Progress reports(feedback) sent till some condition is met.

Learnt about catkin tools and how to create a catkin workspace. After the catkin_make build and devel directories are created. Build contains temporary build files and devel contains the development setup, output etc. The src folder we created can be used to put packages in it. Many build files created are symlinks to the contents of the /opt/ros/noetic

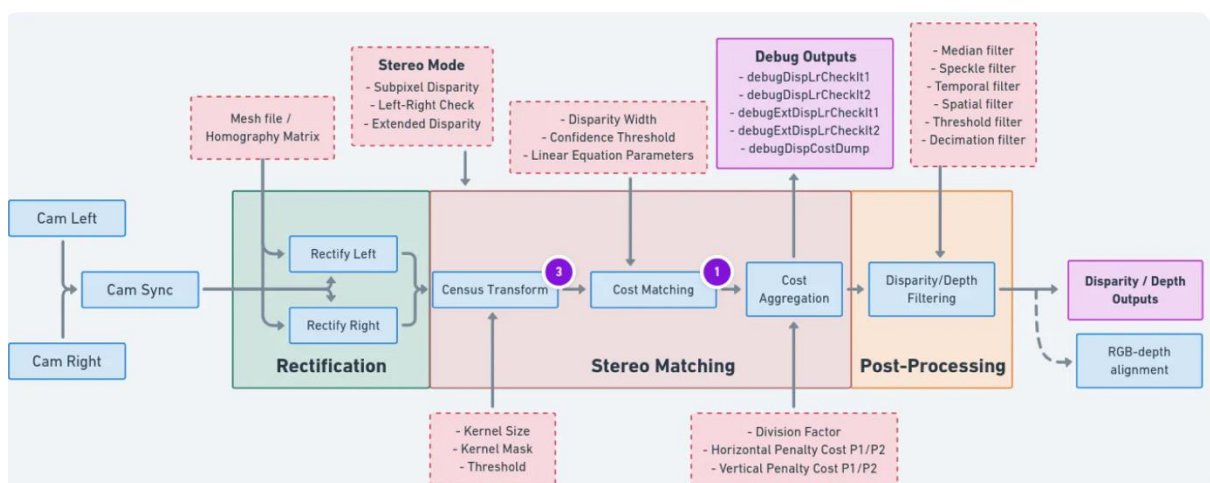
Oak-D:

Pipeline for depth map:

We use stereo vision. Image rectification helps reduce the range to be scanned and disparity is inversely proportional to depth.



Still must go through how the disparity and rectification is calculated.



Finite state machines:

Our auv has 6 degrees of freedom.

Auv jumps between states when some event triggers it. Transition can be thought of as transitioning between states. On jumping to a new state it performs some action (`on_enter()` method in statemachine). `On_exit()` is the opposite method of `on_method()`. My FSM did not account for the rotational movement but only Heave, Surge and Sway.

Note: Forgot to mention about the `mono8` encoding in my comments. This makes sure the output format of conversion is greyscale and 8bit(0-255) pixel variation better suited for ros single pipe. If we wanted to send `rgb` format we would need 3 channels.

```
transmitcanny=bridge.cv2_to_imgmsg(cannyresult, encoding="mono8")
```

```
transmitsobel=bridge.cv2_to_imgmsg(sobelfinal, encoding="mono8")
```