

Inter-IIT Tech Meet 2021

Event name : DRDO Obstacle Avoidance Drone

Submission : ROS package, Documentation

Extra installation :

1. numpy (latest version available)
sudo pip install numpy
2. cv_bridge (latest version available)
3. opencv-python (latest version available)
sudo pip install opencv-contrib-python
4. python 2.7 latest version available)

Parameters Tuned :

fov(depth camera) - 1.3
setpoint_velocity - from LOCAL_NED to BODY_NED in apm_config.yaml
point_cloud_max = 10
point_cloud_min = default value(unchanged)

Running the code :

<source rosnode.sh>

Algorithm :

We are using depth camera input only to command the drone. Whole algorithm works on image processing obtained by analysing the depth camera compressed image and its pixel values. Various conditions are detected by detecting the change in pixels values under different conditions.

1. Using CVBridge to convert the ROS images to OpenCV images.
2. Capture image by the RGB-D, converting all NAN to depth 10m by masking it and cropping the image from top as per needs.
3. Applying binary threshold on the image to identify the free space from the nearby obstacles.
4. Finding the biggest contour and finding its center (cX, cY).

5. The distance between the image centre and contour centre is calculated and the yaw is adjusted accordingly.
6. First yaw is adjusted according to the precision offset set and the drone is aligned with the free space contour. Then it moves 1m forward and then repeats the loop.
7. If there are obstacles very near to the drone a thresholding condition acts and slides the drone opposite to the obstacle density in the image.
8. We check the number of pixels which are less than a given threshold in both left half and right half then slide accordingly.
9. Then we also check the standard deviation of the image to check for dead ends and rotate the drone according to the image. It also adjusts the height.
10. When the world end is reached the aruco marker detection node takes over and lands the drone.

Algorithm For Arucomarker detection-

1. When ArucoMarker enters the field of view of downfacing RGB camera, using opencv aruco detection function, we get the center of aruco marker wrt image coordinates.
2. We know that drone location wrt image is at the center, from these two points we get the unit vector directed to aruco marker center from drone center. By publishing velocity in that unit vector direction, we are able to match the drone center to aruco marker.
3. Then the string is published to the rostopic given accordingly in the problem statement.
4. After getting near the aruco center, by using rosservice we are able to land our drone on the aruco marker center.