APRIL TAGS
PRESENTATION

Mobile Robotics Project

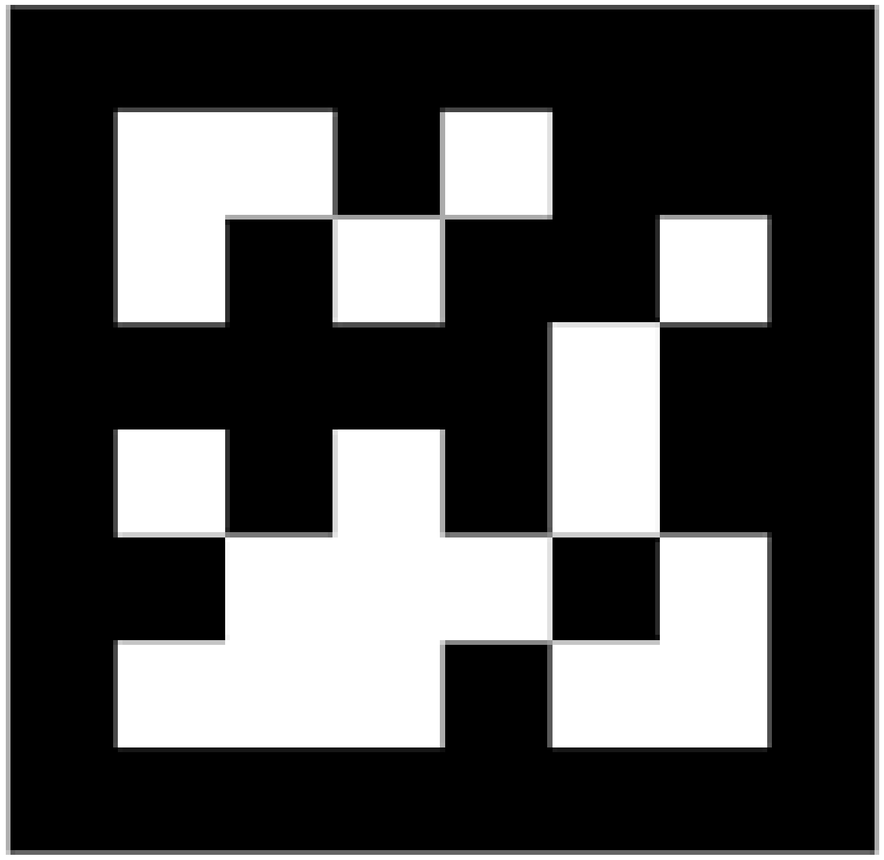
Team - OutOfMemor-

- Anushka Jain
- Himani Belsare
- Surtani Monica

UNDERSTANDING APRIL TAGS

AprilTag is a visual fiducial system, useful for a wide variety of tasks including augmented reality, robotics, and camera calibration.

Targets can be created from an ordinary printer, and the AprilTag detection software computes the precise 3D position, orientation, and identity of the tags relative to the camera.



OUR PROJECT

April Tags based Camera Pose Estimation

HIGHLIGHTS

- Estimate the pose of the camera given the location of April-Tag.
- Consider an environment with multiple April Tags, assuming known location of the robot as it moves build a map of the April Tags.
- Now relocalize in the same environment using the April Tags by observing one or more of the tags and estimate the location from where the Tag is observed

OUR PROGRESS

What have we implemented?

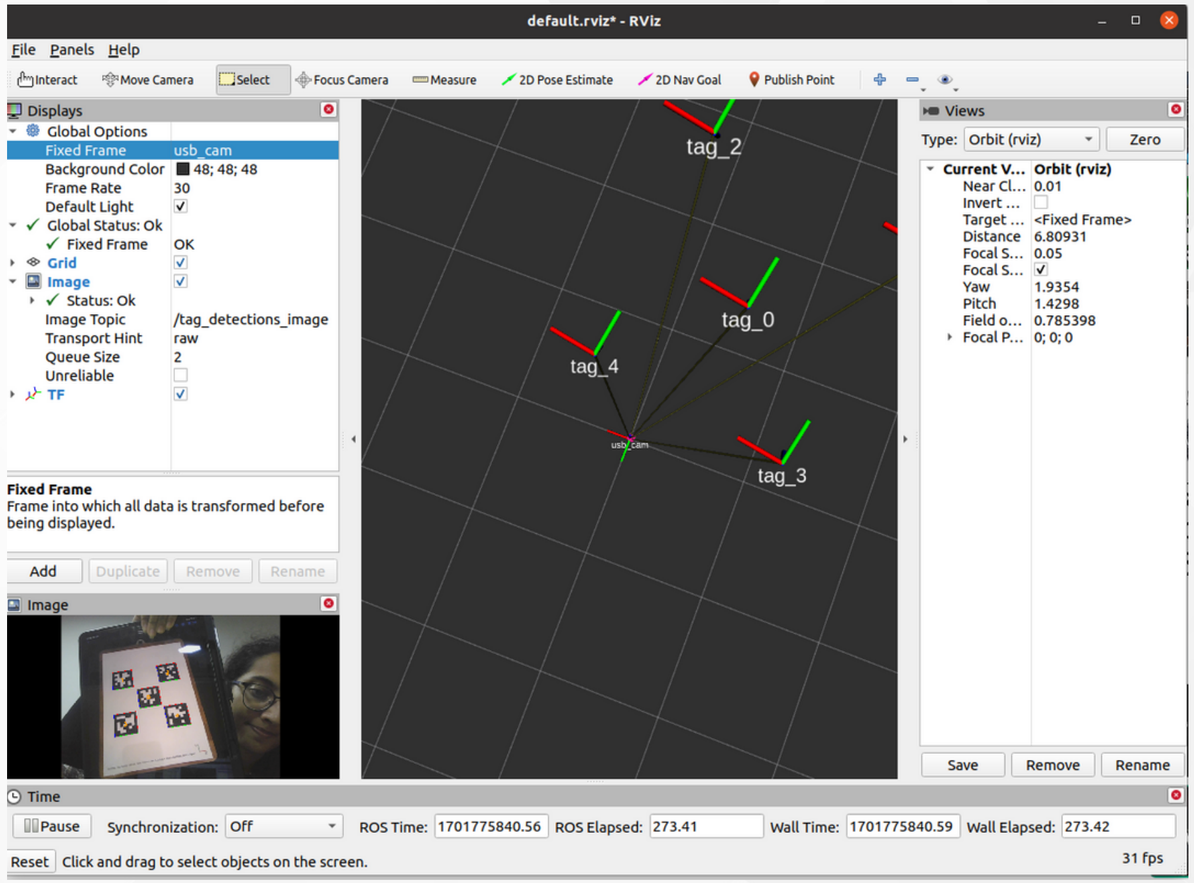
- April Tag detection
- Getting an ID for each detected April Tag
- Camera calibration
- Finding rotation matrix and translation vector
- Using these to:
  - find location of April Tag given known location of the robot
  - find location of robot given location of April Tag
- Using perspective-n-point algorithm
- By using ROS, we have performed camera calibration.
- We are able to detect April tag family and their IDs on ROS using April tag package.

APRIL TAG DETECTION & ID GENERATION

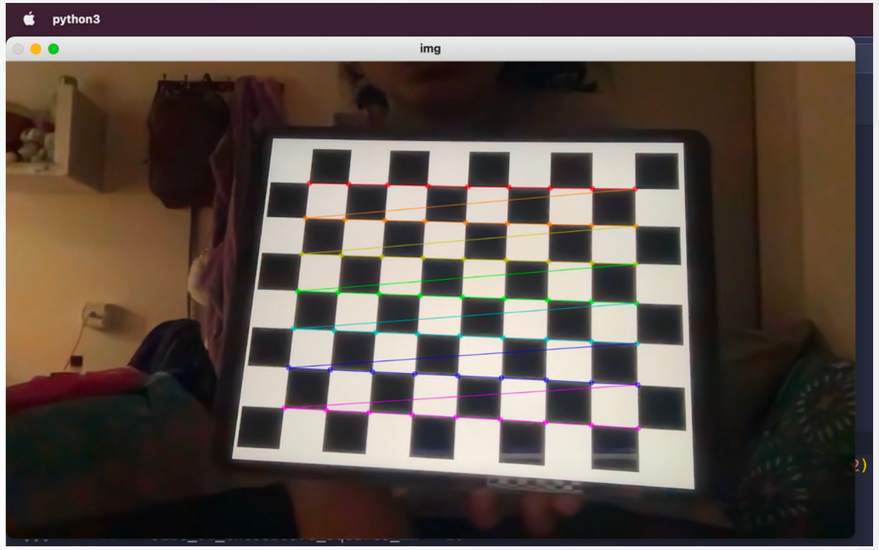
- We used OpenCV and the AprilTags libraries to detect an April Tag Family and ID.
- Using this, different April Tags can be detected and identified as separate entities.
- We are able to detect April tag family and their IDs using April tag package on ROS.

APRIL TAG DETECTION & ID GENERATION

- We are also able to detect April tag family and their IDs using April tag package on ROS.



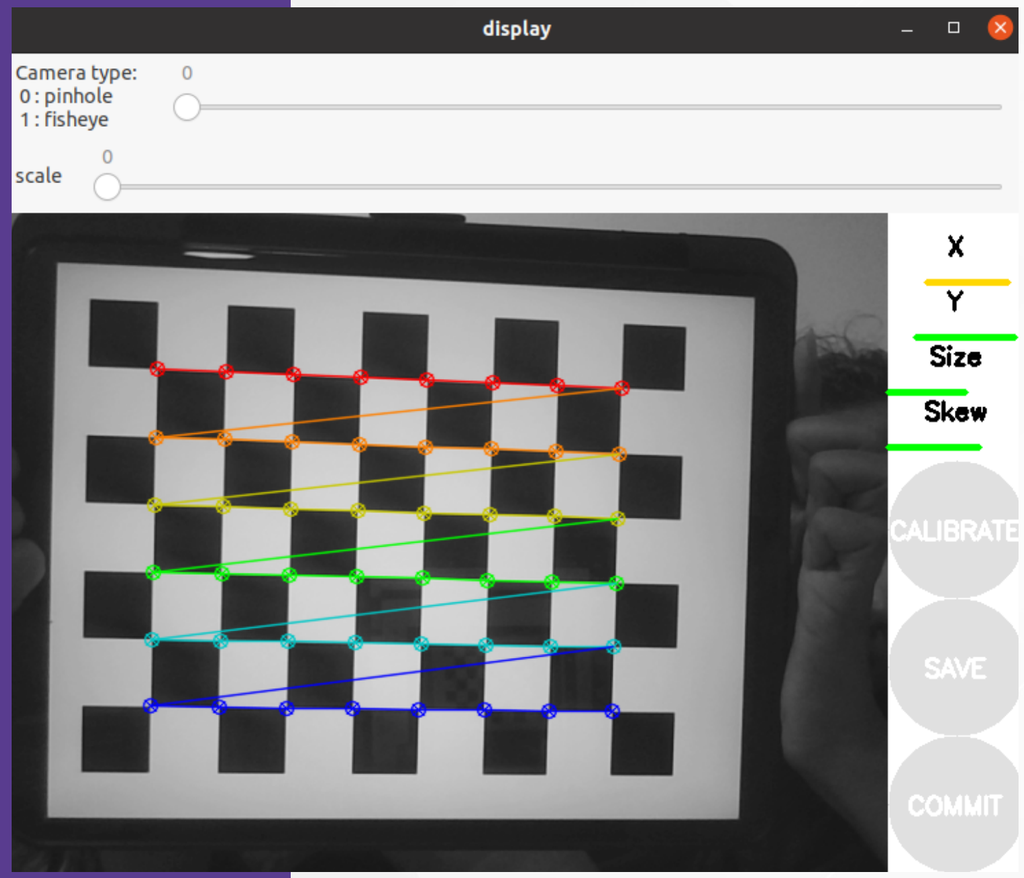
CAMERA CALIBRATION



Using camera calibration to find the following:

- camera matrix - focal length and focal center
- distortion coefficients
- rotation matrix
- translation vector

CAMERA CALIBRATION




```
run_id: 48250072-92c5-11ee-ad8e-095c66bebc56

monica@monica-IdeaPad-3-15IAU7:~/catkin_ws/src/mr$ sudo apt-get install ros-noetic-rqt ros-noetic-rqt
-common-plugins ros-noetic-turtlesim
[sudo] password for monica:
Reading package lists... Done
Building dependency tree
Reading state information... Done
ros-noetic-rqt is already the newest version (0.5.3-1focal.20230620.185852).
ros-noetic-rqt-common-plugins is already the newest version (0.4.9-1focal.20231002.220604).
ros-noetic-turtlesim is already the newest version (0.10.2-1focal.20230620.184400).
ros-noetic-turtlesim set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
monica@monica-IdeaPad-3-15IAU7:~/catkin_ws/src/mr$ rosrun camera_calibration cameracalibrator.py --si
ze 8x6 --square 0.024 image:=/usb_cam/image_raw camera:=/usb_cam
Waiting for service /usb_cam/set_camera_info ...
OK
*** Added sample 1, p_x = 0.482, p_y = 0.459, p_size = 0.364, skew = 0.660
*** Added sample 2, p_x = 0.492, p_y = 0.561, p_size = 0.406, skew = 0.523
*** Added sample 3, p_x = 0.493, p_y = 0.683, p_size = 0.411, skew = 0.427
*** Added sample 4, p_x = 0.578, p_y = 0.792, p_size = 0.437, skew = 0.085
*** Added sample 5, p_x = 0.639, p_y = 0.921, p_size = 0.462, skew = 0.069
*** Added sample 6, p_x = 0.715, p_y = 0.873, p_size = 0.455, skew = 0.158
*** Added sample 7, p_x = 0.696, p_y = 0.900, p_size = 0.422, skew = 0.480
*** Added sample 8, p_x = 0.667, p_y = 0.923, p_size = 0.404, skew = 0.623
*** Added sample 9, p_x = 0.808, p_y = 0.854, p_size = 0.414, skew = 0.072
*** Added sample 10, p_x = 0.798, p_y = 0.702, p_size = 0.397, skew = 0.172
*** Added sample 11, p_x = 0.787, p_y = 0.520, p_size = 0.397, skew = 0.203
*** Added sample 12, p_x = 0.850, p_y = 0.534, p_size = 0.453, skew = 0.135
*** Added sample 13, p_x = 0.873, p_y = 0.593, p_size = 0.456, skew = 0.013
*** Added sample 14, p_x = 0.844, p_y = 0.750, p_size = 0.481, skew = 0.017
*** Added sample 15, p_x = 0.785, p_y = 0.295, p_size = 0.479, skew = 0.221
*** Added sample 16, p_x = 0.350, p_y = 0.371, p_size = 0.451, skew = 0.095
*** Added sample 17, p_x = 0.283, p_y = 0.316, p_size = 0.438, skew = 0.007
*** Added sample 18, p_x = 0.306, p_y = 0.295, p_size = 0.405, skew = 0.179
*** Added sample 19, p_x = 0.356, p_y = 0.235, p_size = 0.395, skew = 0.087
*** Added sample 20, p_x = 0.463, p_y = 0.437, p_size = 0.474, skew = 0.113
*** Added sample 21, p_x = 0.377, p_y = 0.299, p_size = 0.532, skew = 0.134
*** Added sample 22, p_x = 0.467, p_y = 0.382, p_size = 0.489, skew = 0.253
*** Added sample 23, p_x = 0.398, p_y = 0.455, p_size = 0.507, skew = 0.200
*** Added sample 24, p_x = 0.346, p_y = 0.567, p_size = 0.546, skew = 0.209
*** Added sample 25, p_x = 0.280, p_y = 0.401, p_size = 0.522, skew = 0.060
*** Added sample 26, p_x = 0.438, p_y = 0.293, p_size = 0.495, skew = 0.015
*** Added sample 27, p_x = 0.629, p_y = 0.305, p_size = 0.489, skew = 0.018
*** Added sample 28, p_x = 0.659, p_y = 0.416, p_size = 0.524, skew = 0.049
*** Added sample 29, p_x = 0.636, p_y = 0.584, p_size = 0.552, skew = 0.033
*** Added sample 30, p_x = 0.603, p_y = 0.715, p_size = 0.562, skew = 0.006
*** Added sample 31, p_x = 0.435, p_y = 0.688, p_size = 0.570, skew = 0.005
*** Added sample 32, p_x = 0.483, p_y = 0.600, p_size = 0.530, skew = 0.044
*** Added sample 33, p_x = 0.370, p_y = 0.543, p_size = 0.504, skew = 0.049
*** Added sample 34, p_x = 0.345, p_y = 0.205, p_size = 0.541, skew = 0.062
```

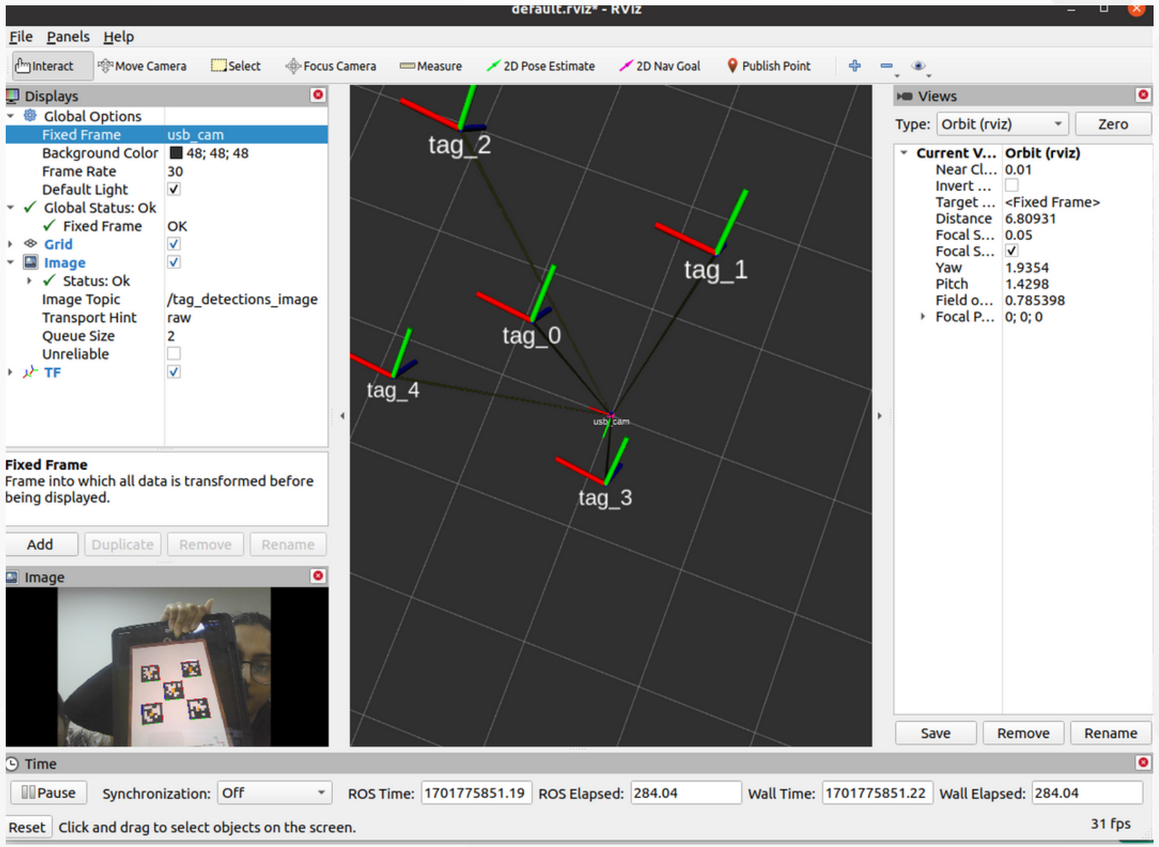
FINDING ROTATION AND TRANSLATION

Using OpenCV and the April Tags libraries, we are able to obtain the rotation and translation matrix to transform points from the object co-ordinate system to the robot co-ordinate system.

These matrices can be used to perform further calculations to get more information regarding positions of entities in the environment.

DETECTION OF APRIL TAGS ON ROS

Using ROS and the April Tag package, we are able to obtain the April tag family and their IDs on screen while the April tags are infront of camera.



THANK YOU