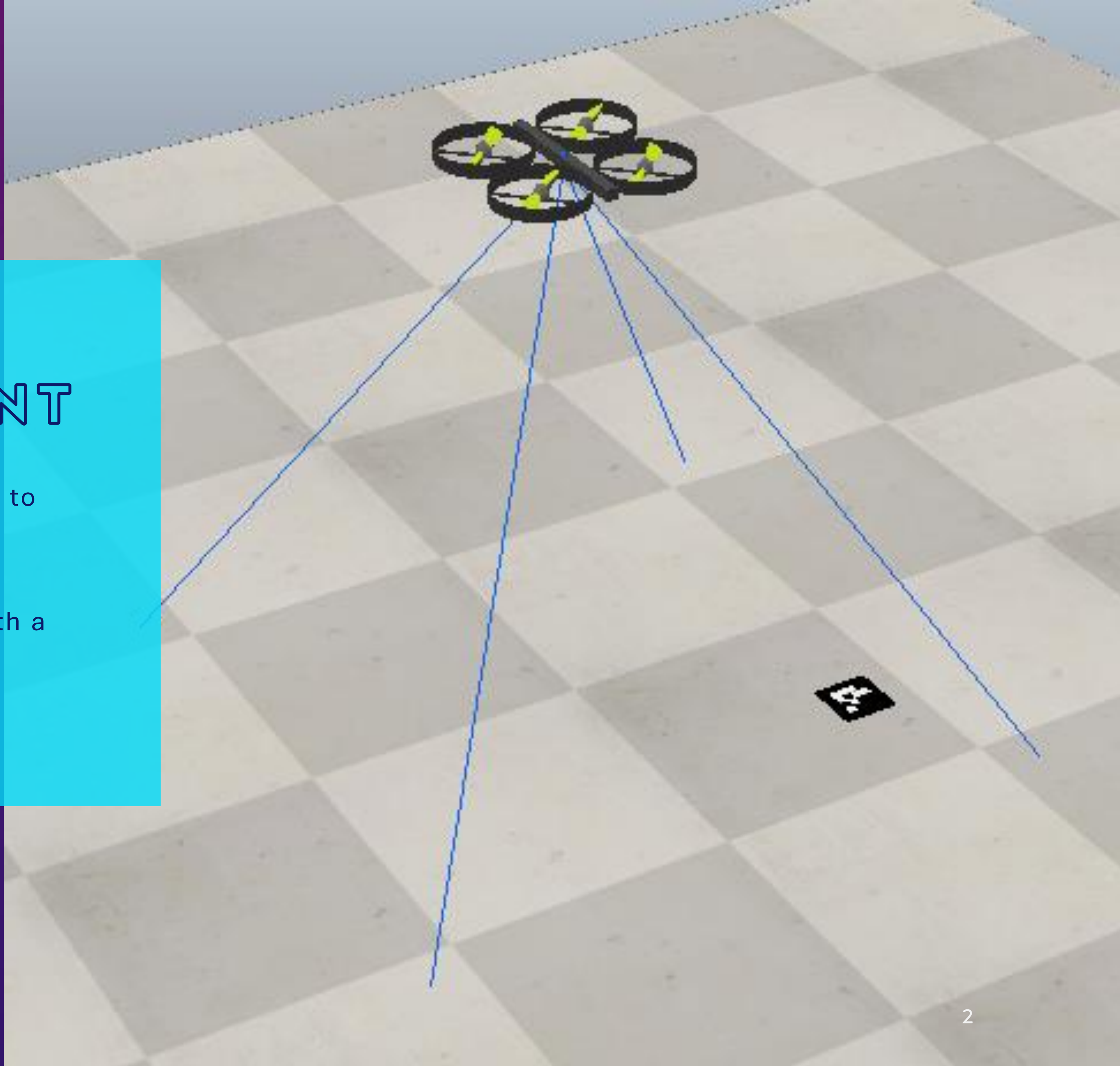
MARKER TRACKING WITH DRONE

Aravind Seshadri
aravinds21@iitk.ac.in

PROBLEM STATEMENT

Marker Tracking has various applications related to Search and Rescue, Surveying and Mapping, etc. The aim is to develop a reliable and efficient tracking system for ArUco markers and test it with a Tello drone.

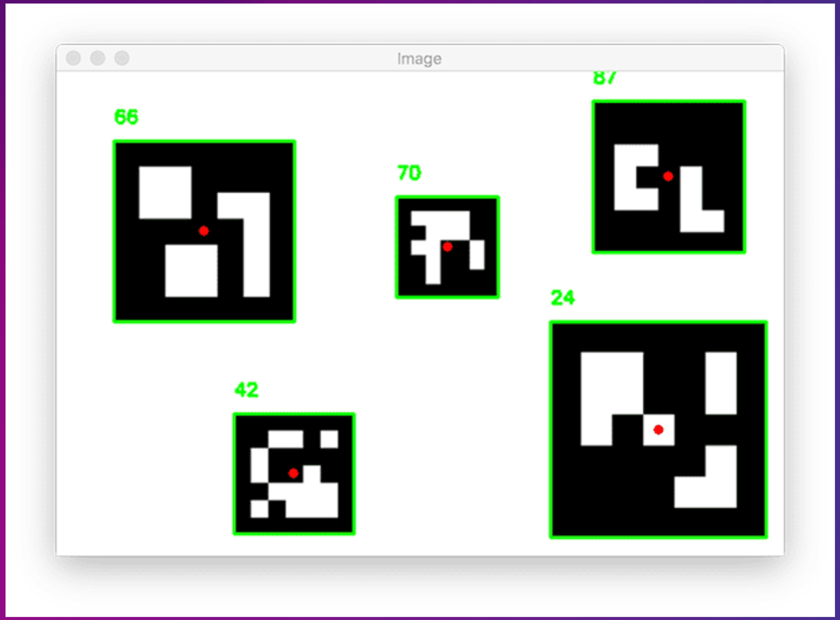


REPOSITORY

https://drive.google.com/drive/folders/1_ISxxmItlG7TvLwKUBkKO0RX3jpLSh27?usp=sharing

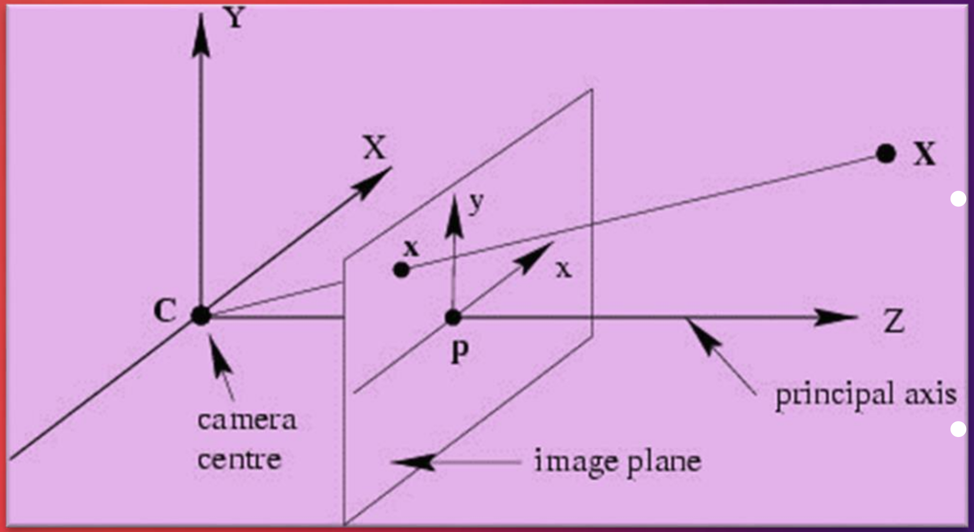
METHOD TO SOLVE THE PROBLEM

- Images are obtained from the drone using a camera
- OpenCV has a library for ArUco marker detection
- The library includes certain functions such as detect.markers() and pose_estimation()



TRACKING USING OPENCV

- Once the marker is detected on the image, its actual x,y,z positions with respect to the camera are calculated.
- Various methods are present to calculate these parameters.
- In this case, the pose_estimation function returns the rotation and translation vectors of the marker with respect to the camera.



- We require the intrinsic camera parameters to accurately calculate the positions, which can be obtained by camera calibration.
- This includes the focal length of the camera. Using the method of triangulation, the positions can be calculated.
- Once the positions are calculated, an effective control algorithm can position the drone along the center of the marker.

CONTRIBUTIONS

SIMULATIONS

- The initial phase included simulating various environments for the drone.
- An API was established between MATLAB and Coppleliasim to control the drone from MATLAB.
- A PID controller was designed to enable the drone to hover or fly to any other given position.

- Using the MexOpenCV interface, the Aruco marker library was imported into MATLAB. The detect_marker function returned the corners of the marker in the image.

- A basic control algorithm was designed using the error between the center of the marker and that of the image.
- This is used to position the drone above the marker.

Implementation with the Tello drone

- To implement the tracking algorithms in the Tello Drone, first, the DJITelloPy API reference was understood.
- The library was imported into Python, and the basic functionalities of the drone were experimented with.

- These include tello.forward(),tello.up(),etc and the tello.send_rc_control() command, which can be used to set the drone's speed for Horizontal, Vertical and Angular Motion.

- The aruco marker libraries are imported, and the rotation, and translation matrices are obtained using pose estimation after camera calibration

FURTHER READING

Additionally, methods to calculate the coordinates of an object from the image were studied.
For depth information, some of the methods included:

- Method of triangulation: Using a know distance and given width in the image, the focal length can be calculated. With this information and using the property of similarity between triangles, the distance can be calculated.
- Stereo vision is a much more effective way to calculate the distance
- Using the size of the marker and intrinsic properties of the camera, coordinates can be calculated.

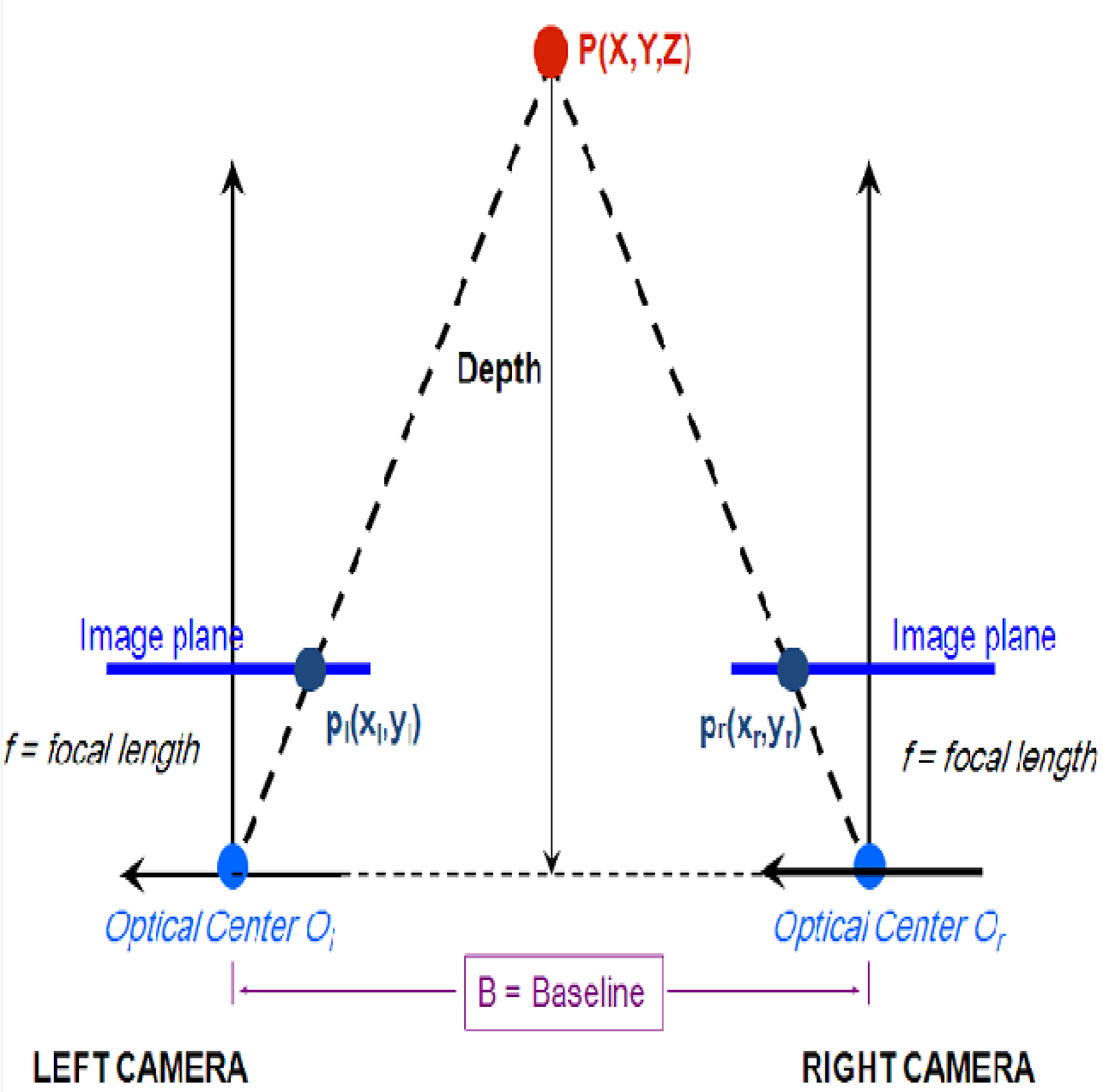
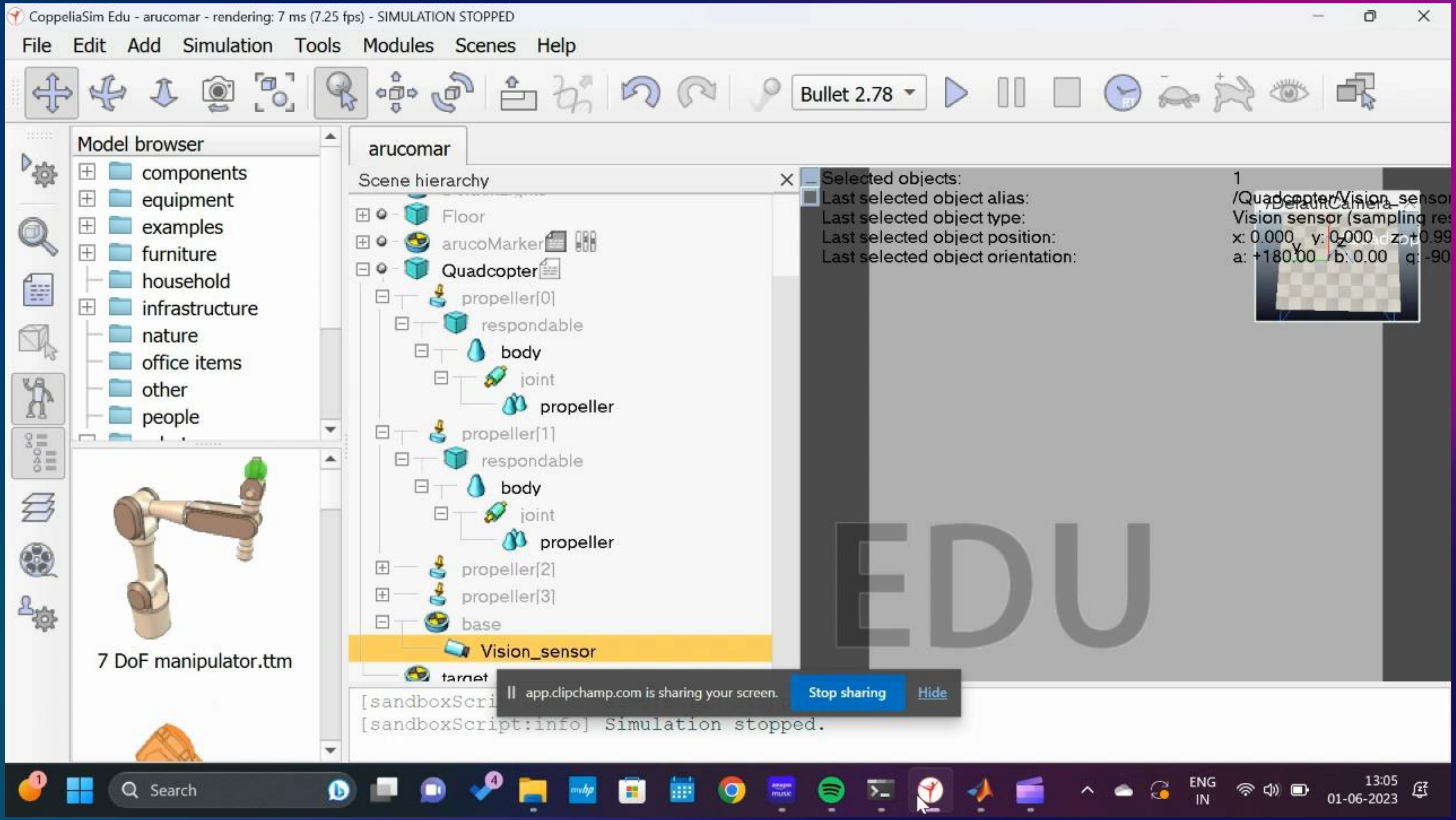


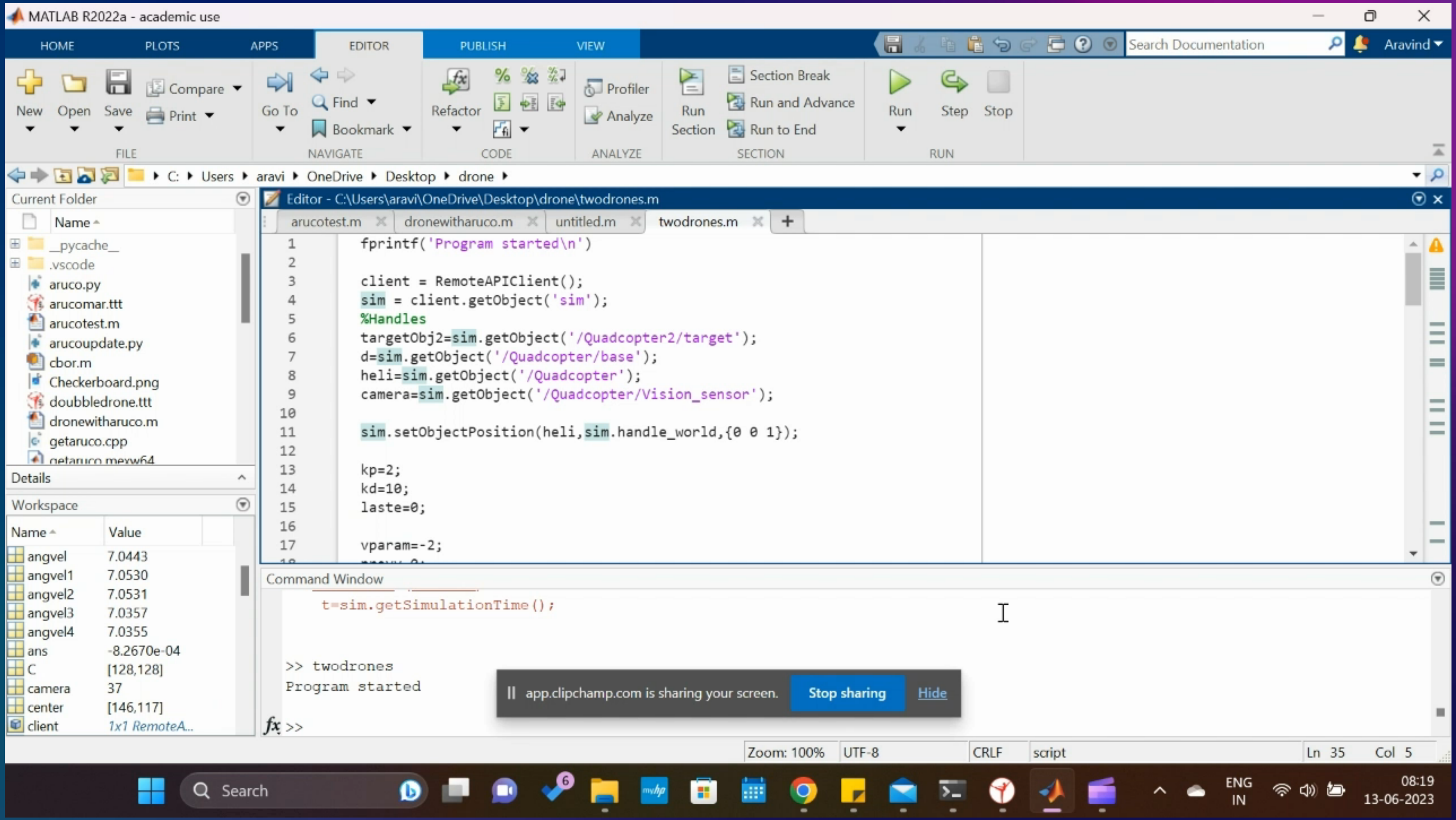
Fig. 2. Stereo vision basics

DRONE LAND

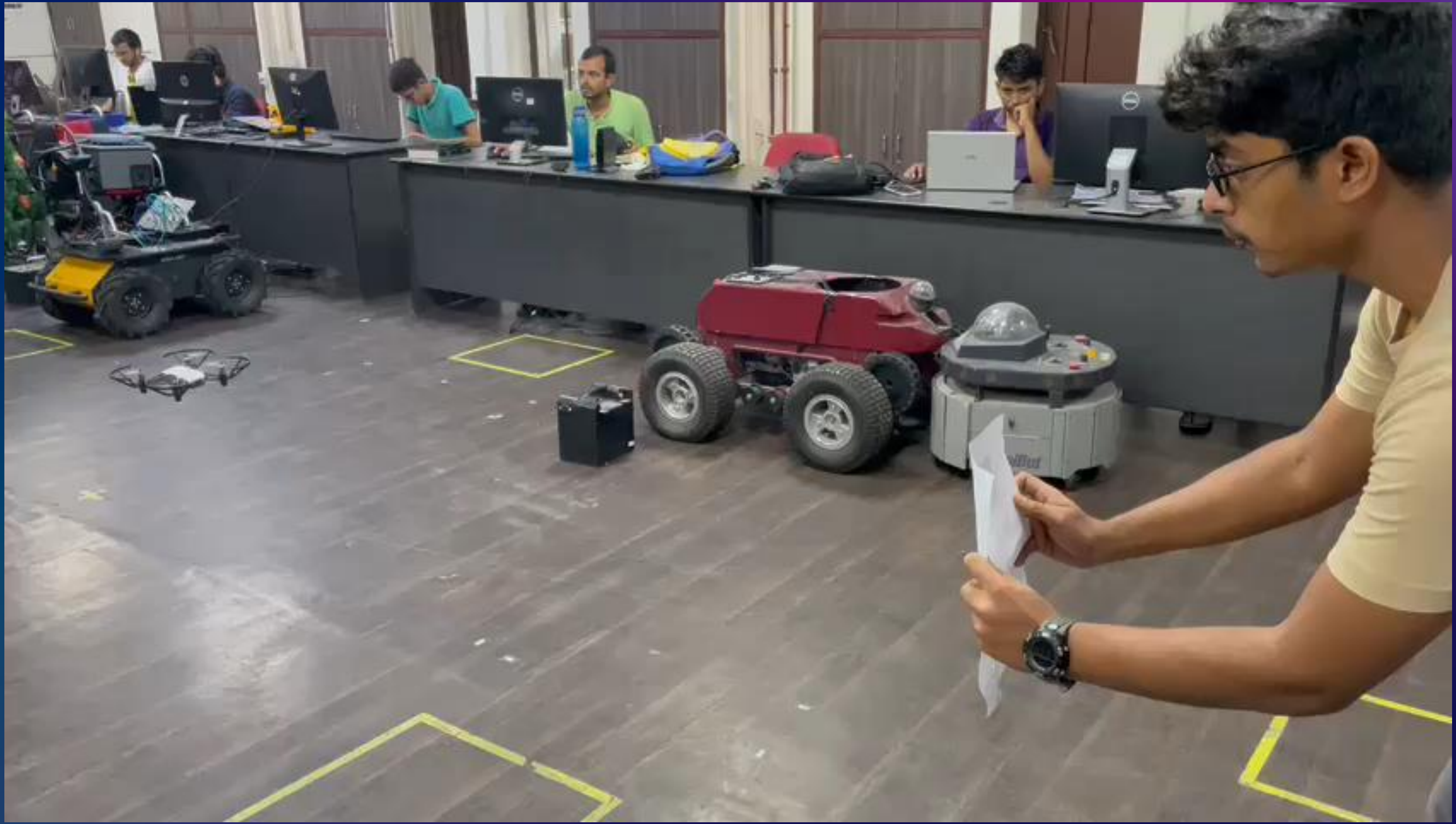


CoppeliaSim Edu - arucomar - rendering: 7 ms (7.25 fps) - SIMULATION STOPPED
File Edit Add Simulation Tools Modules Scenes Help
Bullet 2.78
Model browser
components
equipment
examples
furniture
household
infrastructure
nature
office items
other
people
7 DoF manipulator.ttm
arucomar
Scene hierarchy
Floor
arucoMarker
Quadcopter
propeller[0]
respondable
body
joint
propeller
propeller[1]
respondable
body
joint
propeller
propeller[2]
propeller[3]
base
Vision_sensor
target
Selected objects: 1
Last selected object alias:
Last selected object type:
Last selected object position:
Last selected object orientation:
EDU
app.clipchamp.com is sharing your screen.
Stop sharing
Hide
[sandboxScript:info] Simulation stopped.
Search
ENG
IN
13:05
01-06-2023

DRONE FOLLOW



TELLO IMPLEMENTATION



NEXT TASK

Develop better control over the drone and increase the stability of the system.