ENPM 809T

UMCP, Mitchell, Summer 2019

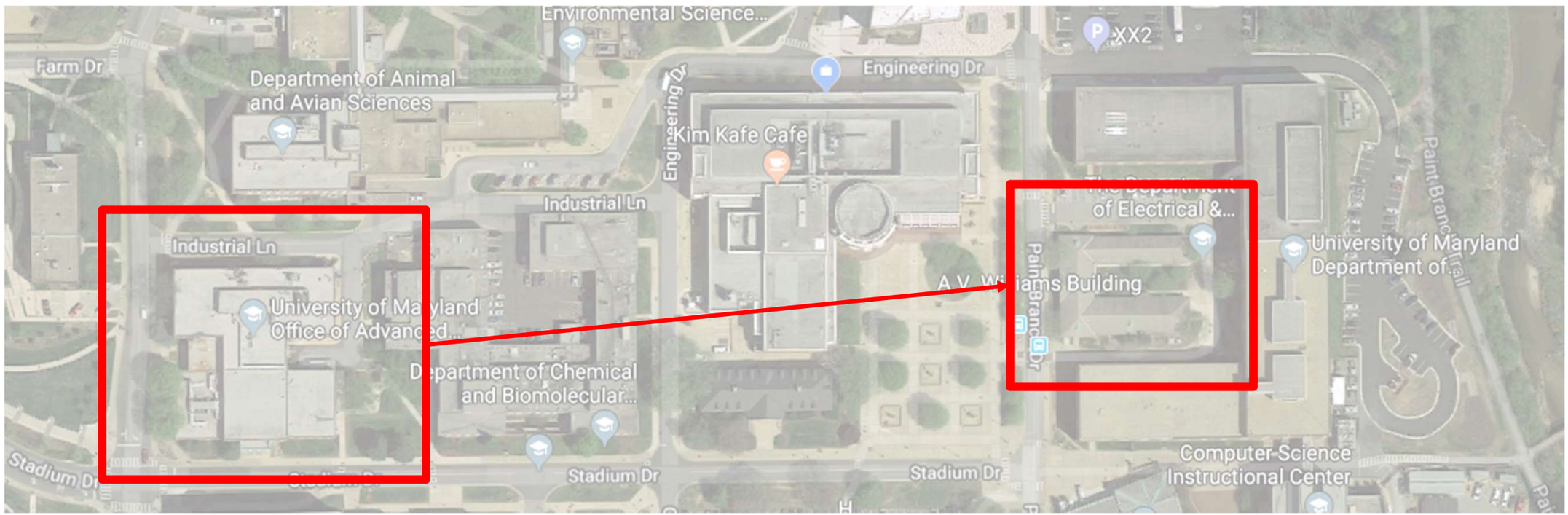
Robotics Realization Lab (RRL)

- Engineering Annex
- Middle of AV Williams Building

https://sites.google.com/site/roboticsrealizationlab/

Robotics Realization Lab (RRL)

- Engineering Annex
- Middle of AV Williams Building



https://sites.google.com/site/roboticsrealizationlab/

Robotics Realization Lab (RRL)

https://sites.google.com/site/roboticsrealizationlab/

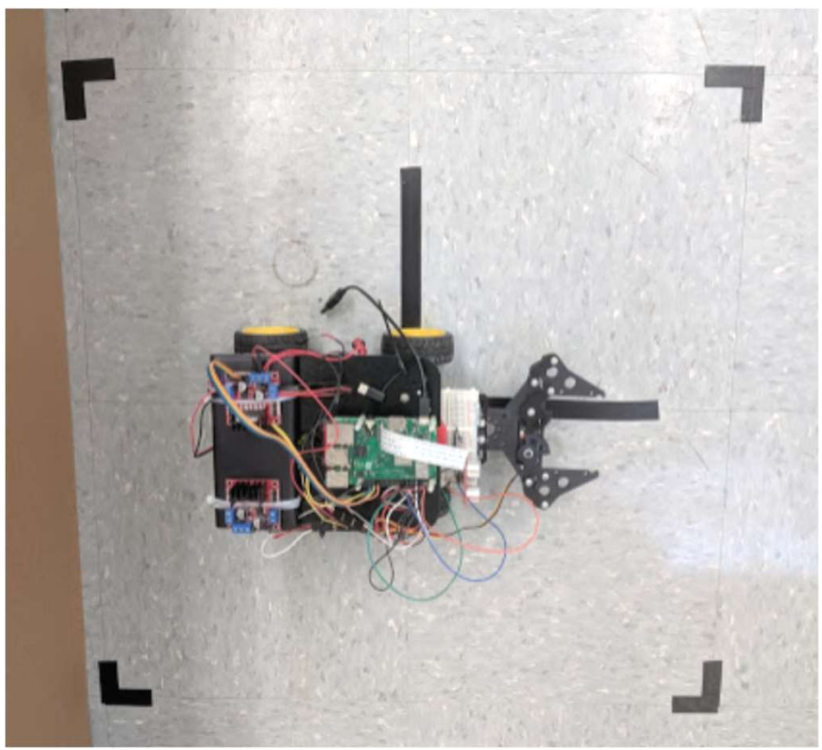
Robotics Realization Lab (RRL)

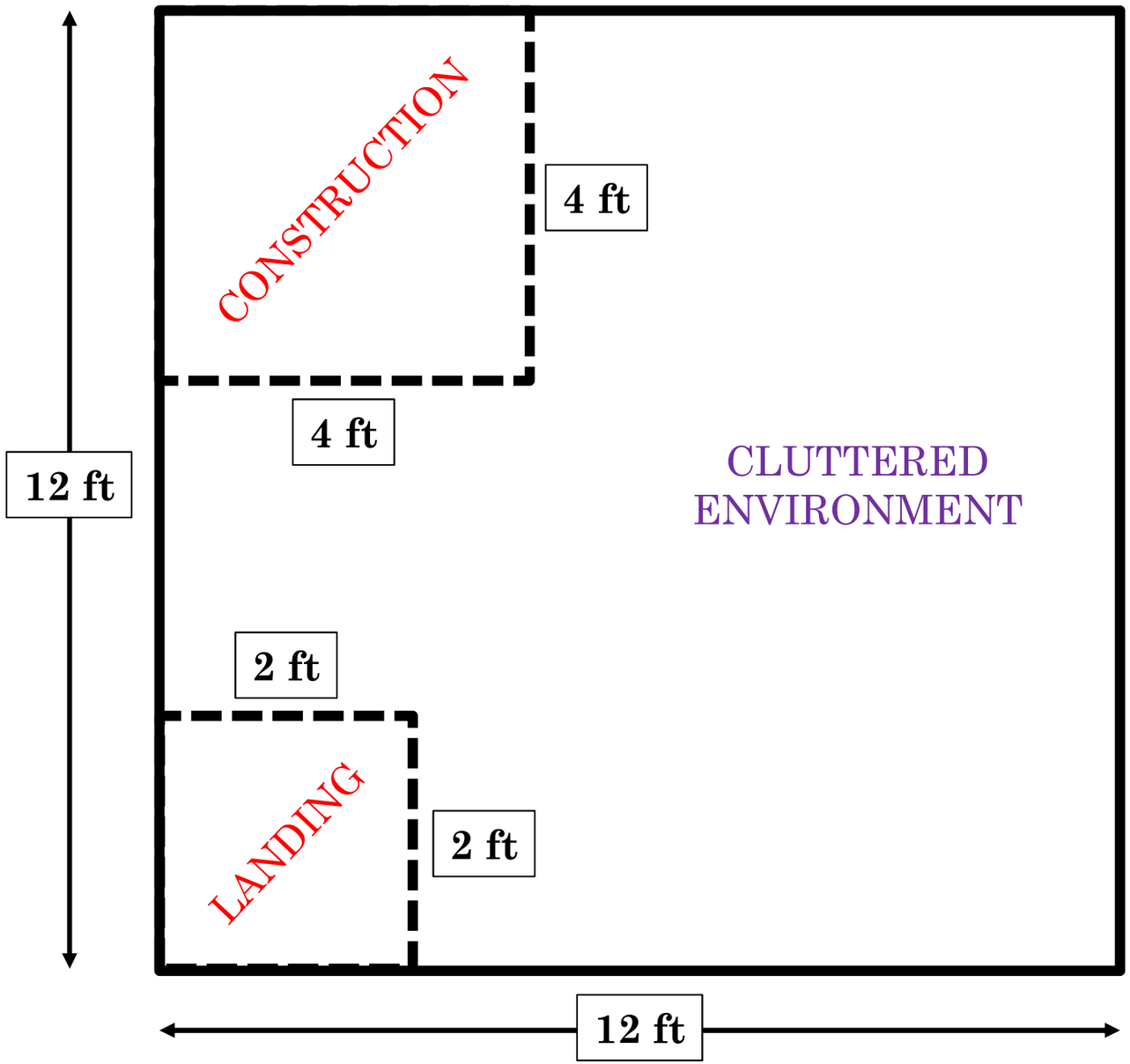
https://sites.google.com/site/roboticsrealizationlab/

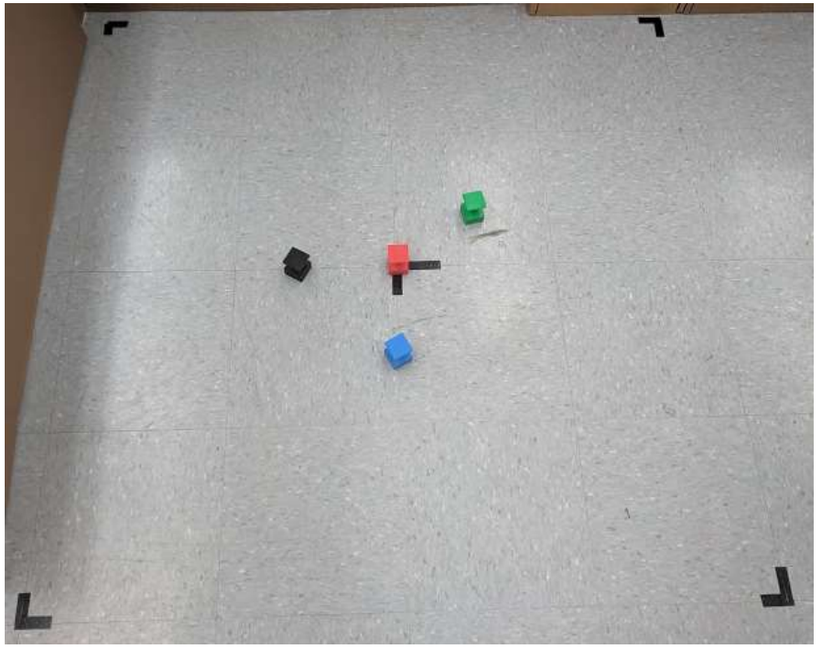
Robotics Realization Lab (RRL)

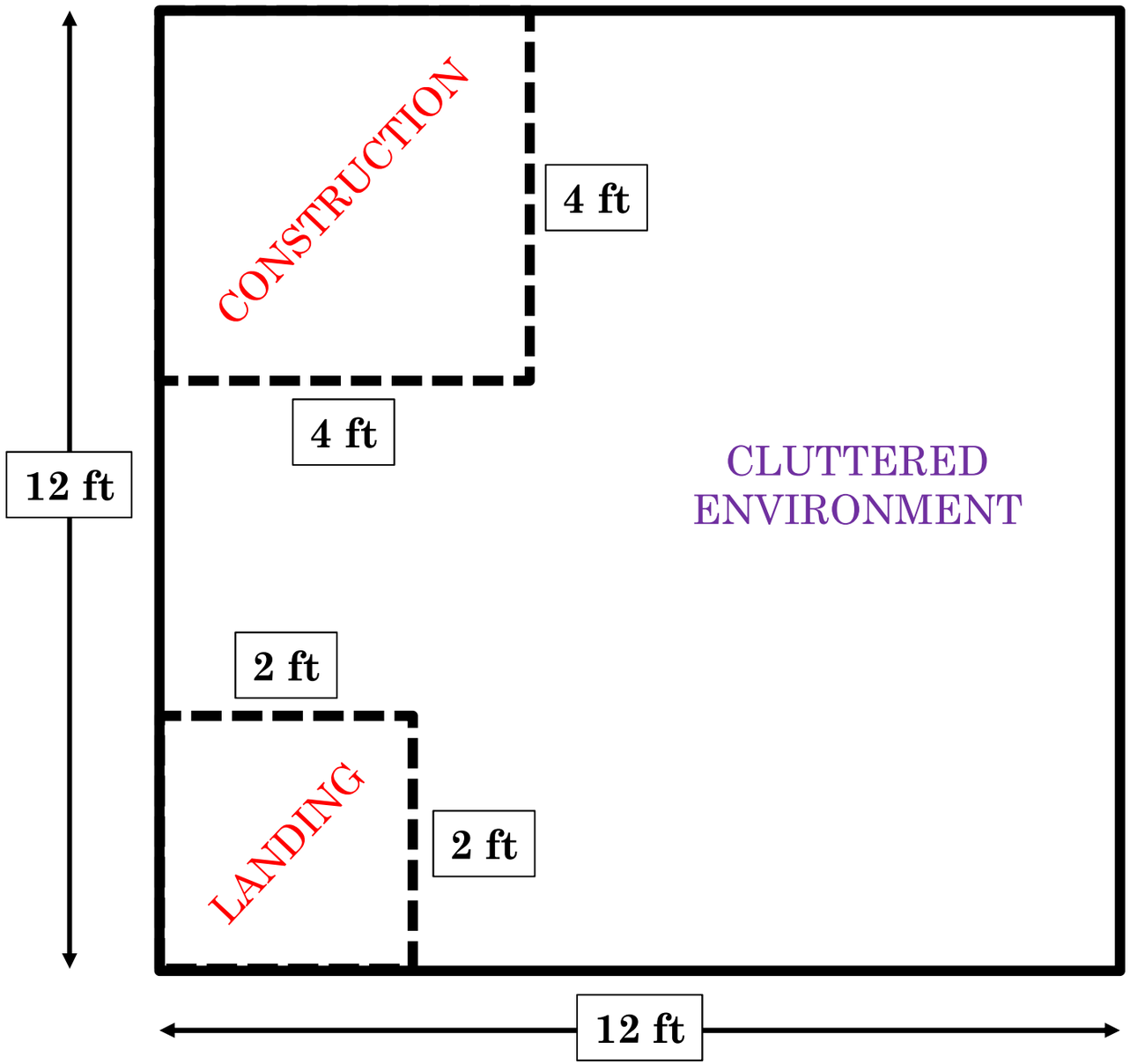
- Tuesday July 30th, 5:30 - 8:45pm
- Tuesday August 6th, 5:30 - 8:45pm

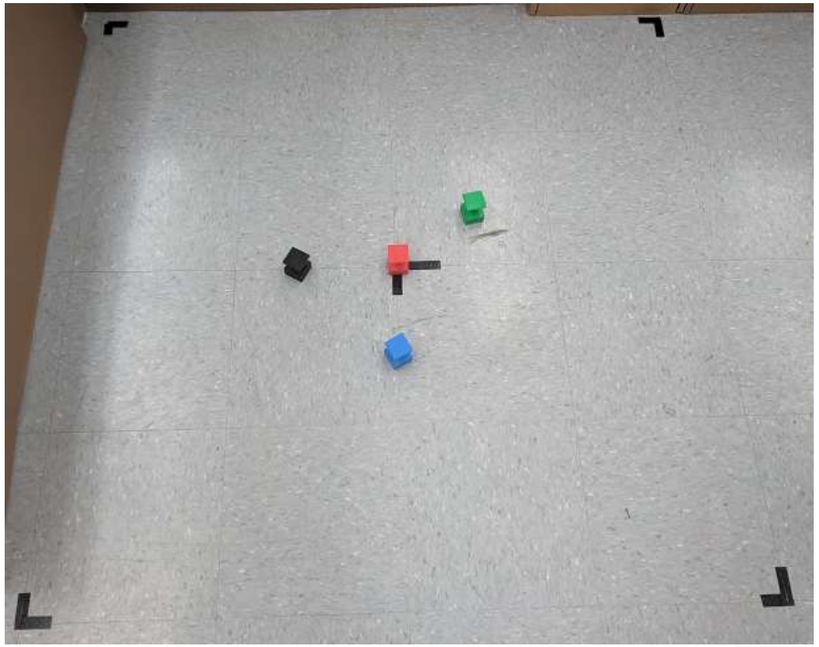

Physics-Aware Planning for
Autonomous Unmanned Surface Vehicles
30
strataSYS

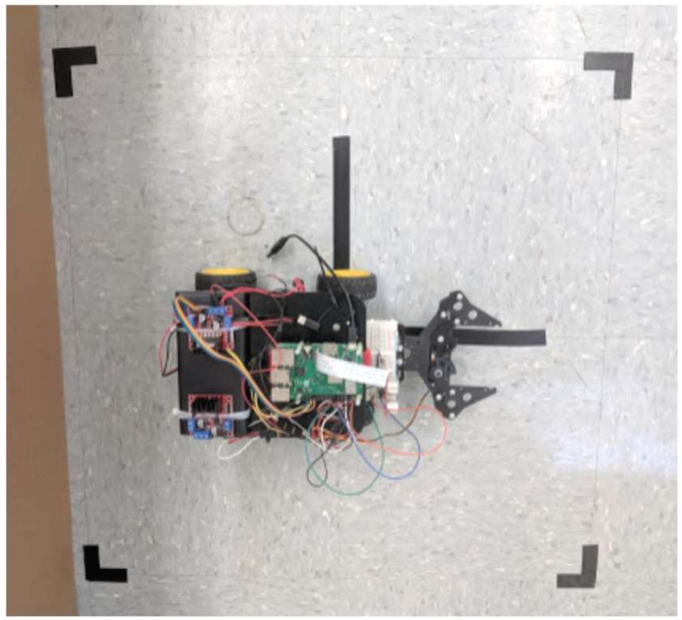


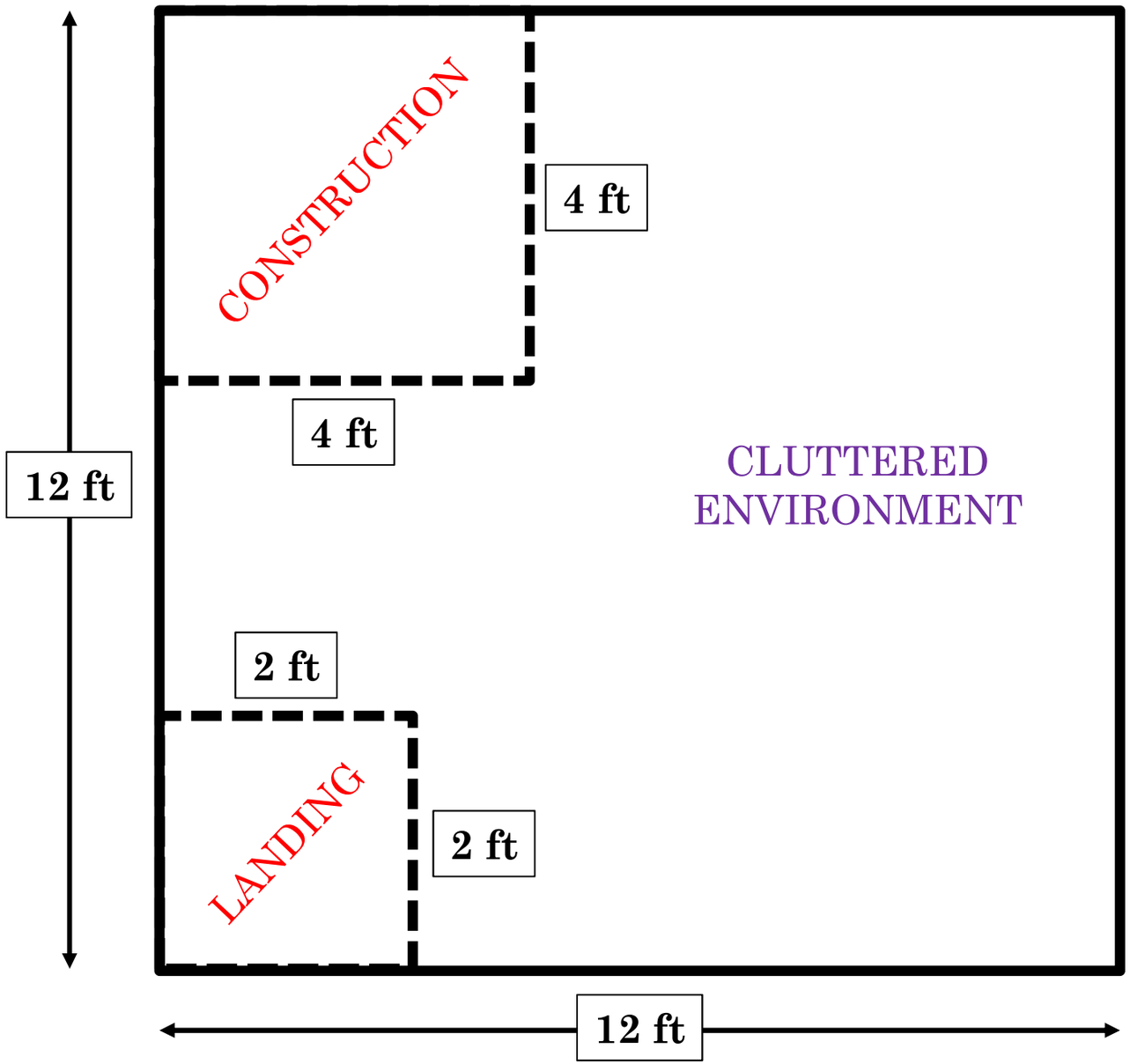

CONSTRUCTION
4 ft
4 ft
12 ft
CLUTTERED
ENVIRONMENT
2 ft
LANDING
2 ft
12 ft

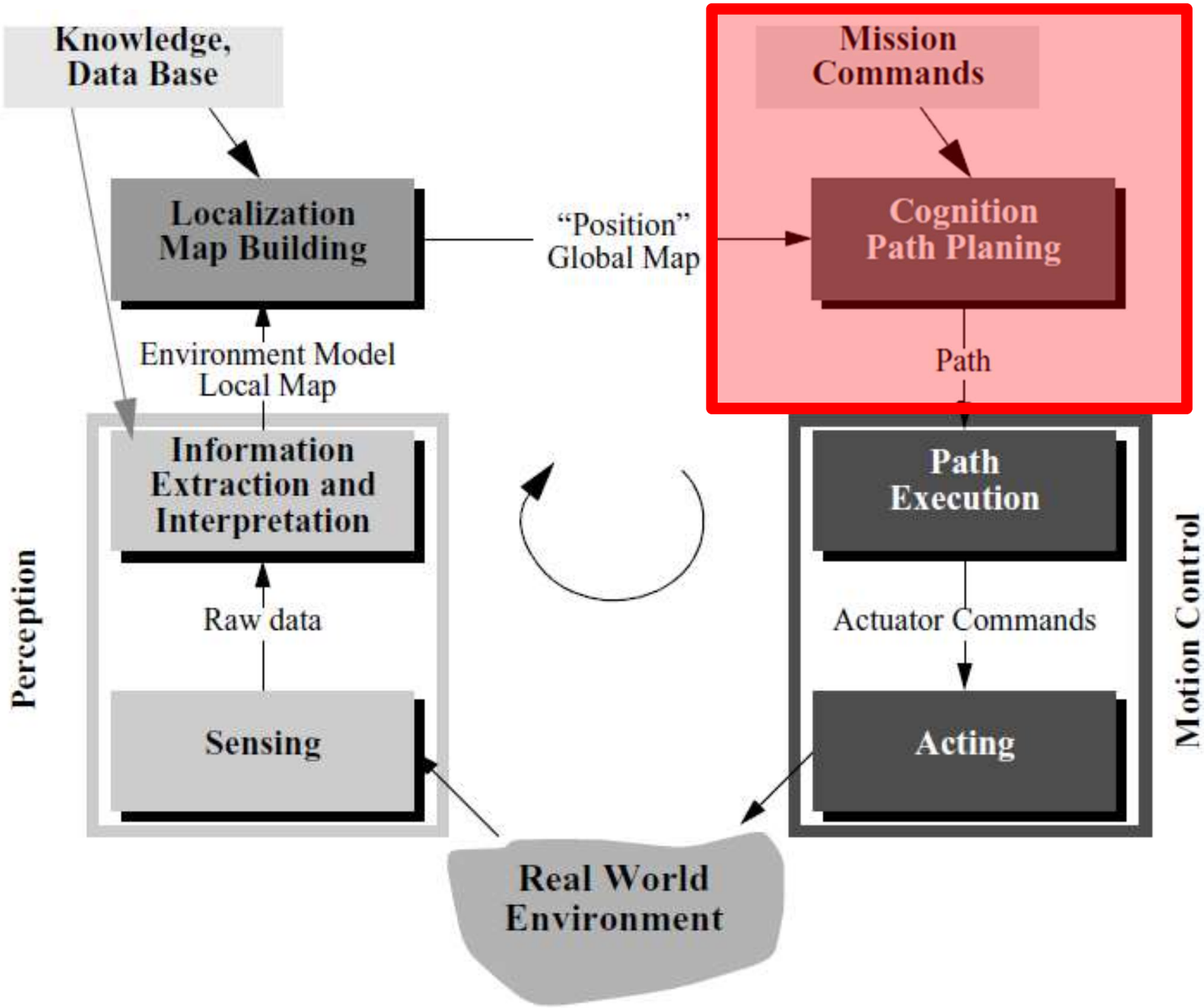



CONSTRUCTION
4 ft
4 ft
12 ft
CLUTTERED
ENVIRONMENT
2 ft
LANDING
2 ft
12 ft



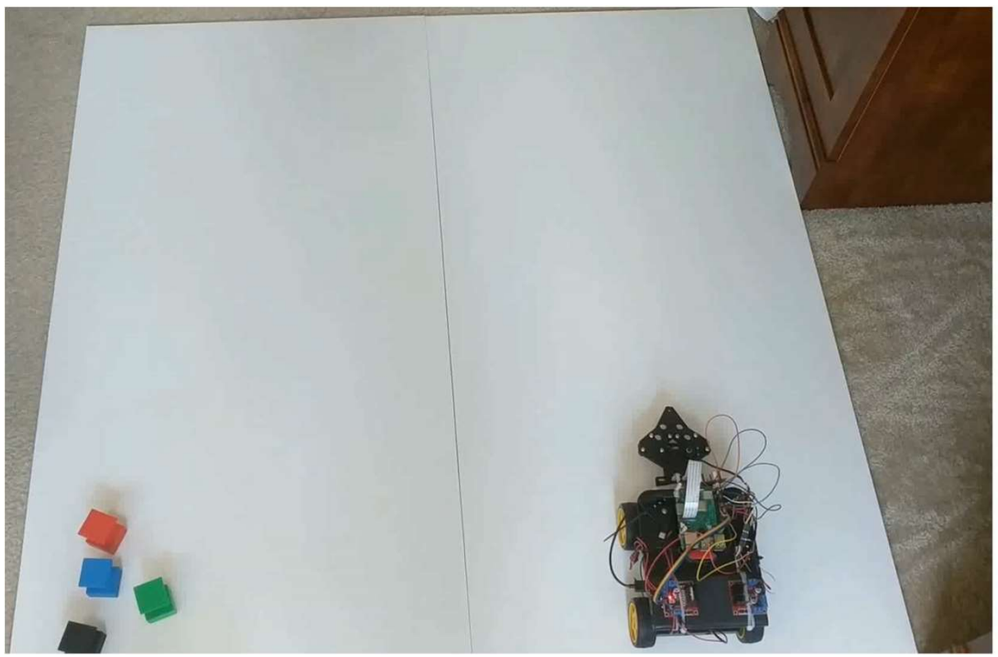


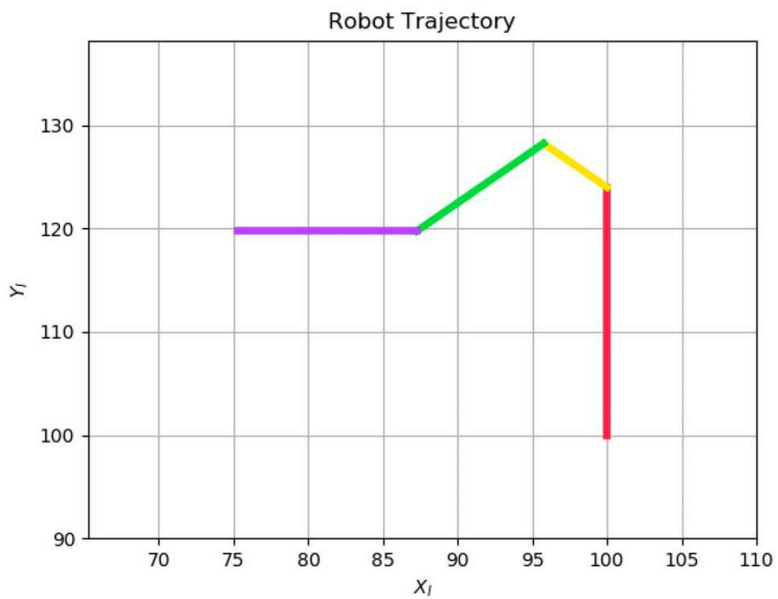

CONSTRUCTION
4 ft
4 ft
12 ft
CLUTTERED
ENVIRONMENT
2 ft
LANDING
2 ft
12 ft


Knowledge,
Data Base
Mission
Commands
Localization
Map Building
"Position"
Global Map
Cognition
Path Planing
Environment Model
Local Map
Path
Information
Extraction and
Interpretation
Path
Execution
Perception
Raw data
Actuator Commands
Motion Control
Sensing
Acting
Real World
Environment

In-Class Exercise

- Referring to HW #8, demonstrate your robot:
  1. Take as input a sequence of commands from operator
  2. Drive robot through sequence
  3. Record position data through sequence
- Once complete, open & plot position data in Matplotlib





Robot Functionality

- Perception

1. RPi camera (***picamera***, ***raspistill***, ***raspivid***)
2. Ultrasonic range sensor (***range01.py***, ***drive01.py***)

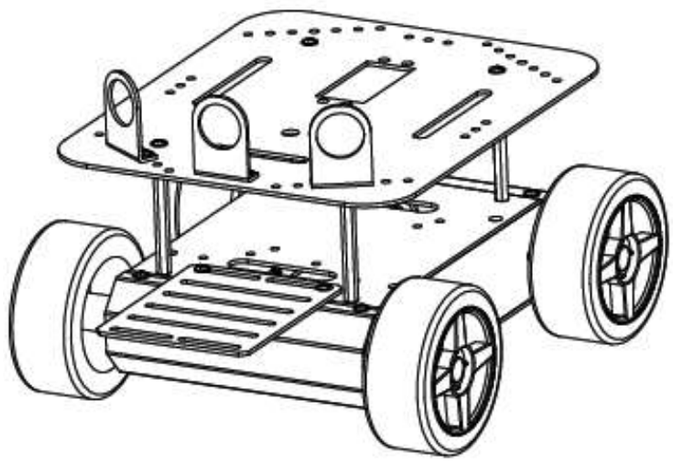
- Locomotion

1. H-bridge (***motorcontrol01.py***)
2. Servo gripper (***servocontrol01.py***)

- Localization

1. Motor encoders (***encodercontrol01.py***, ***map01.py***)
2. Email communication (***email01.py***)

- Planning & Navigation

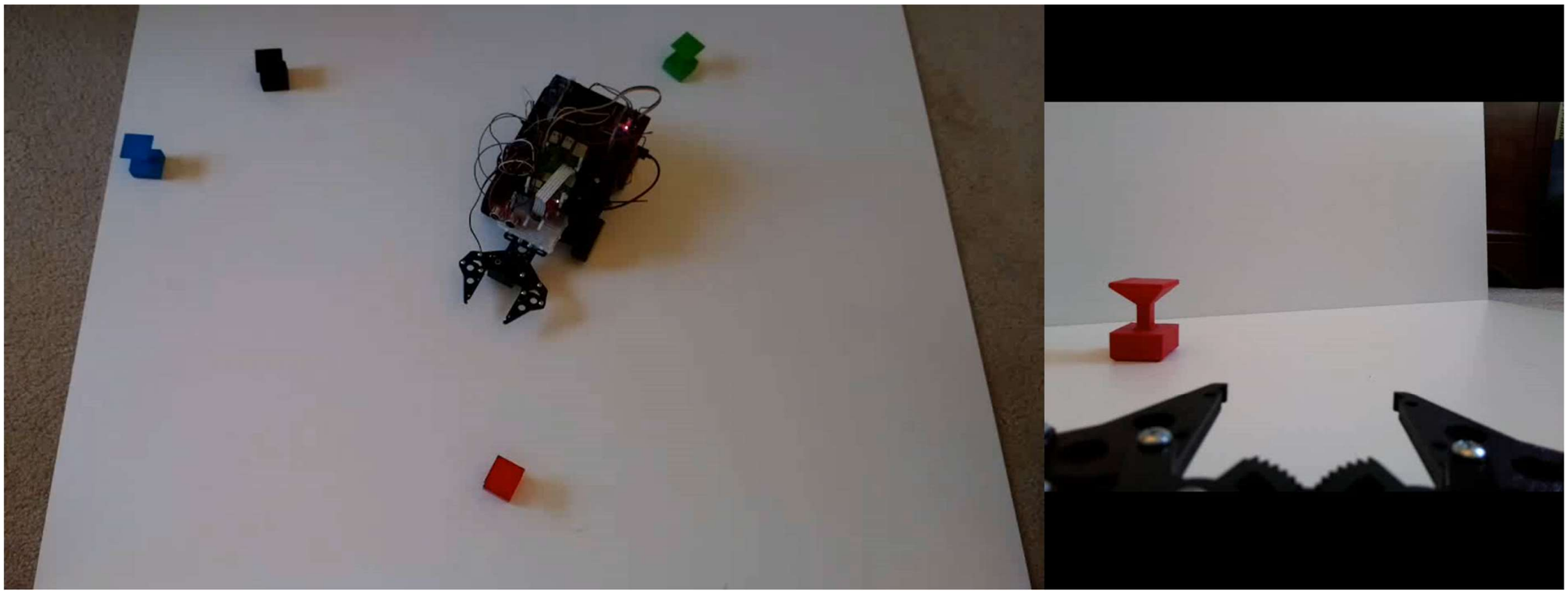


Planning & Navigation

- We now transition to our robot's cognitive level
- **Cognition**: the purposeful decision-making & execution that a system utilizes to achieve its highest-order goals
- The specific aspect of cognition directly linked to robust robot mobility is **navigation competence**
- Path planning & obstacle avoidance

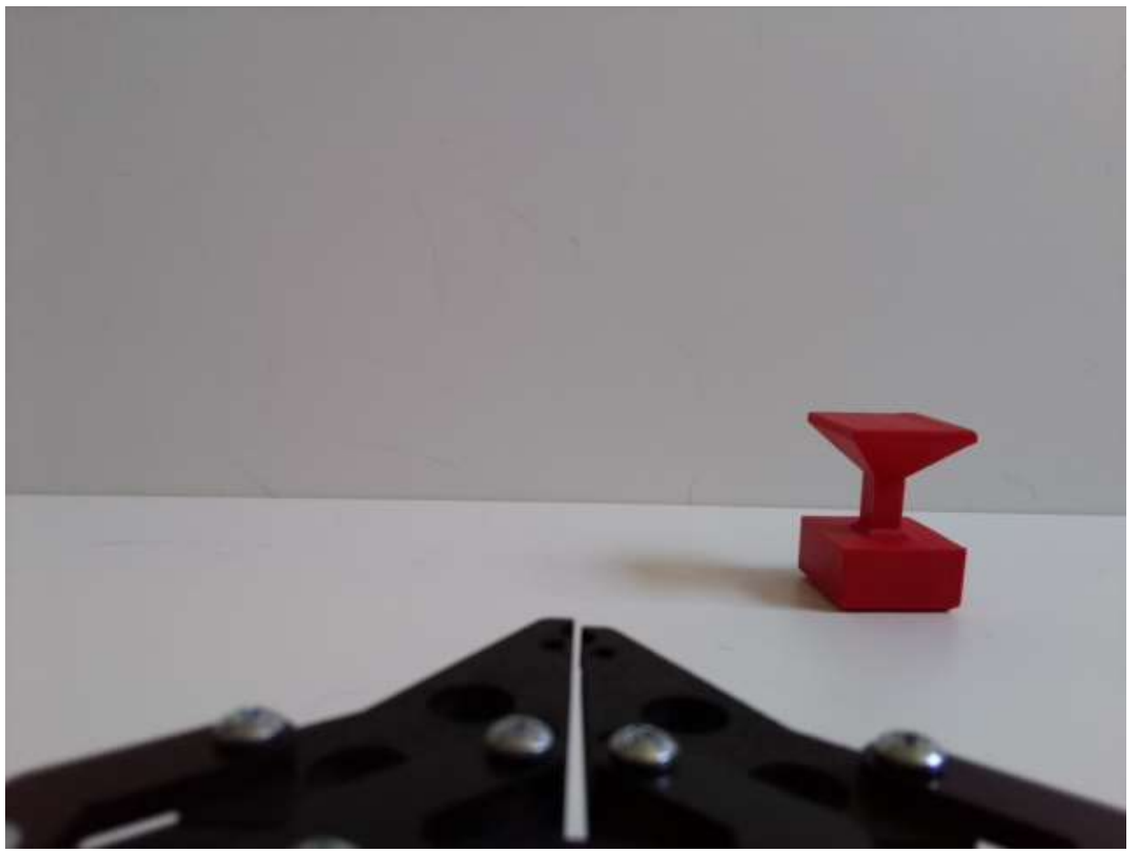
Planning & Navigation

- First step: **autonomously** track moving object



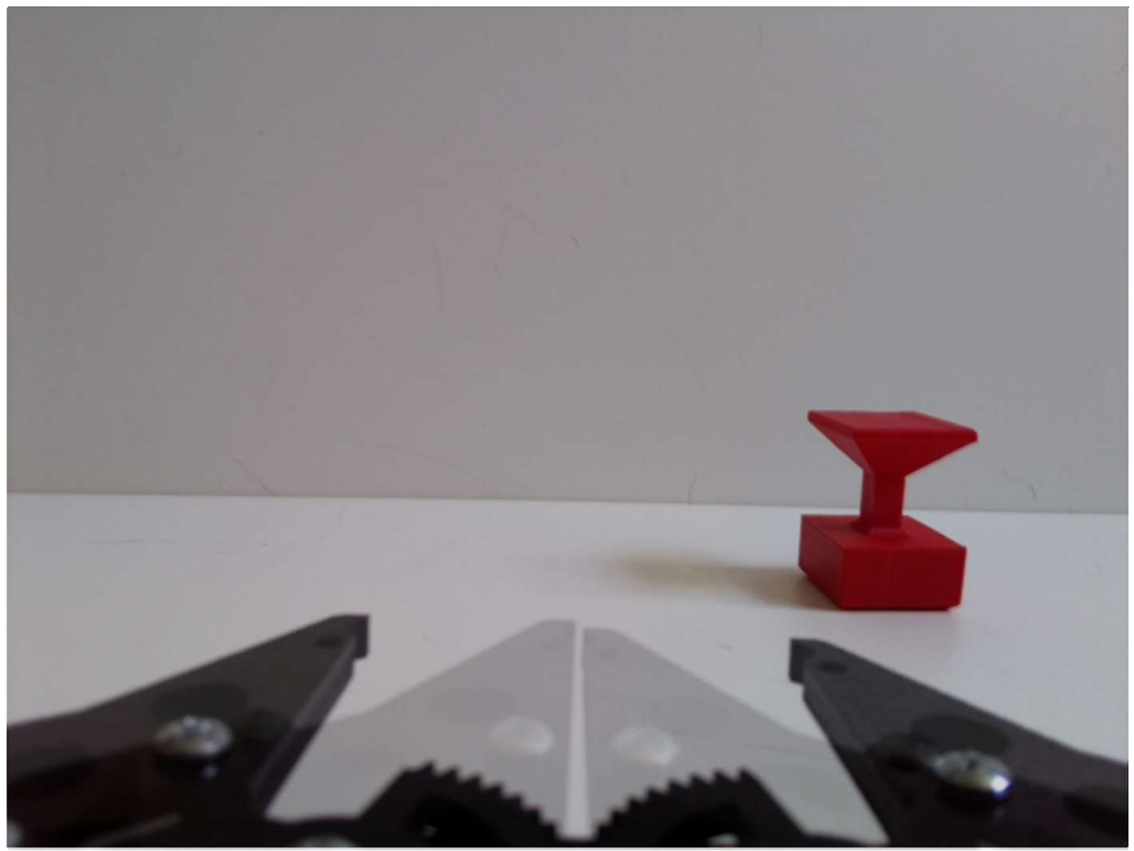
Planning & Navigation

- Begin by setting servo gripper to open position



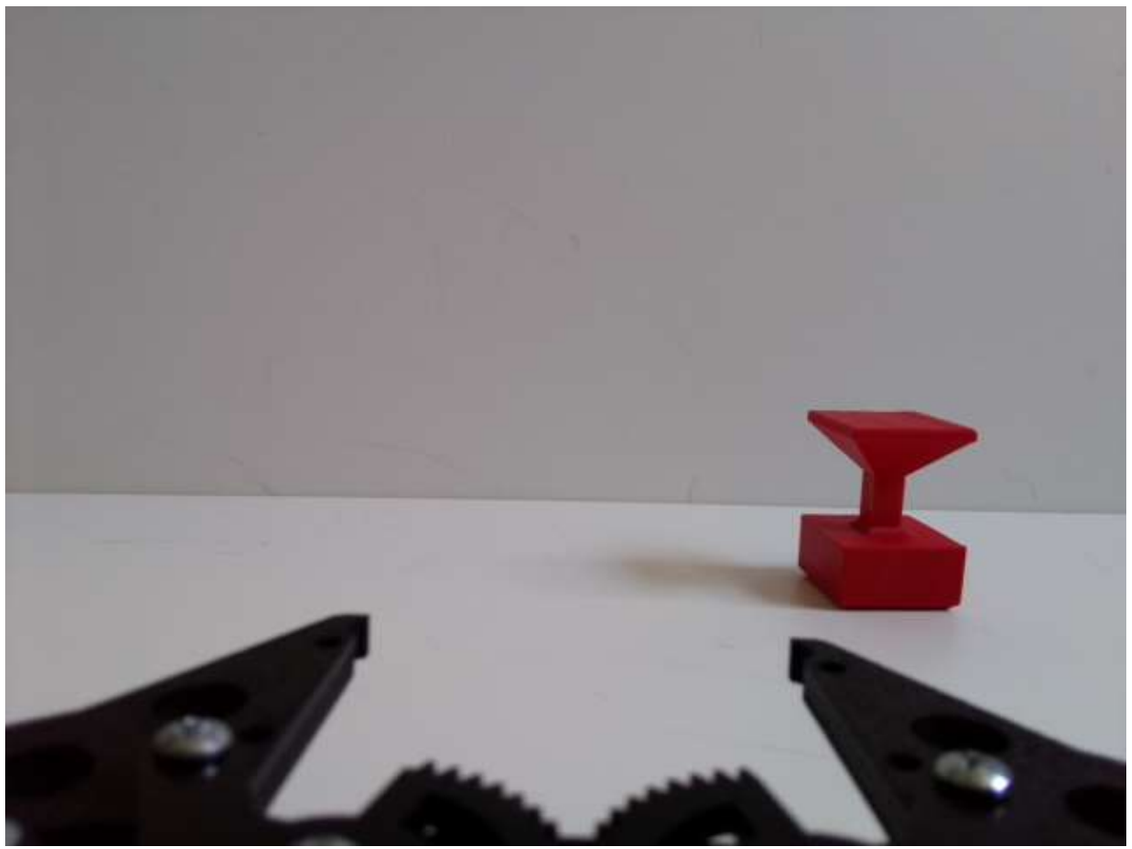
Planning & Navigation

- Begin by setting servo gripper to open position



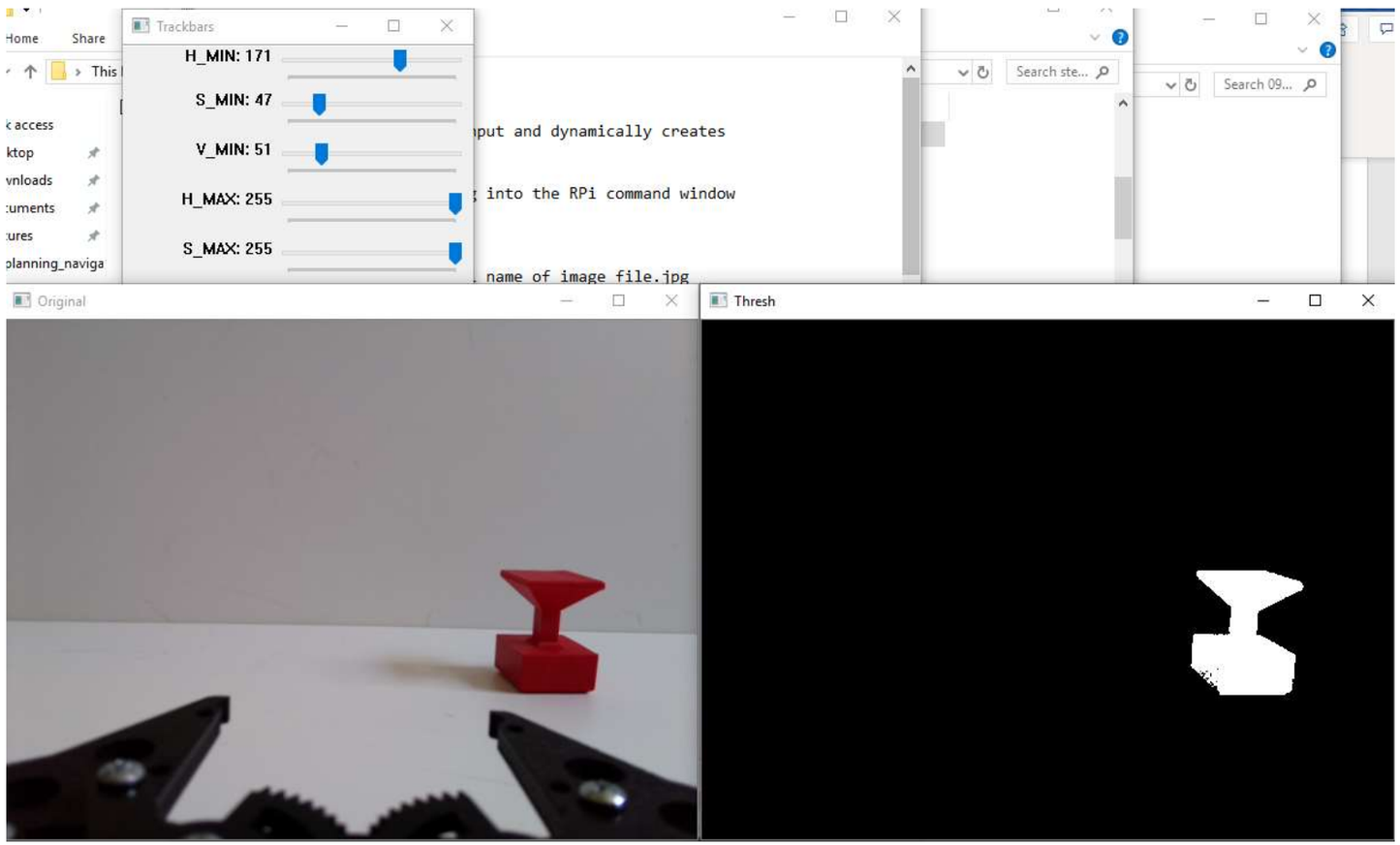
Planning & Navigation

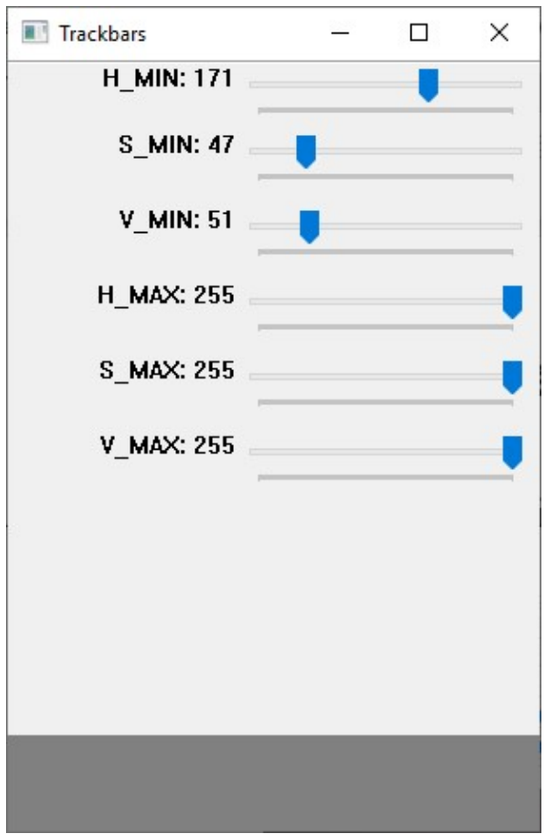
- Use RPi camera to record 640x480 image of the scene



Planning & Navigation

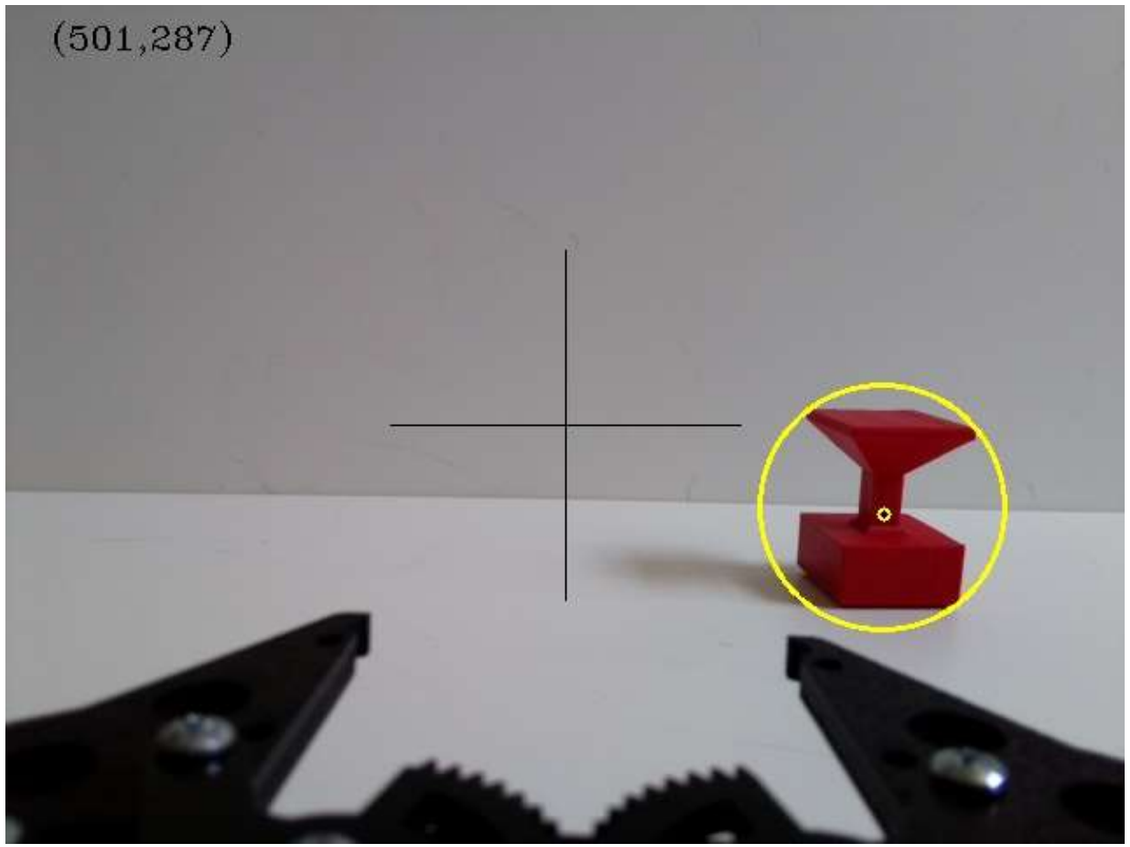
- Use ***colorpicker.py*** to determine HSV color bounds





Planning & Navigation

- Refer to HW #3: apply mask(s) & find center of contour



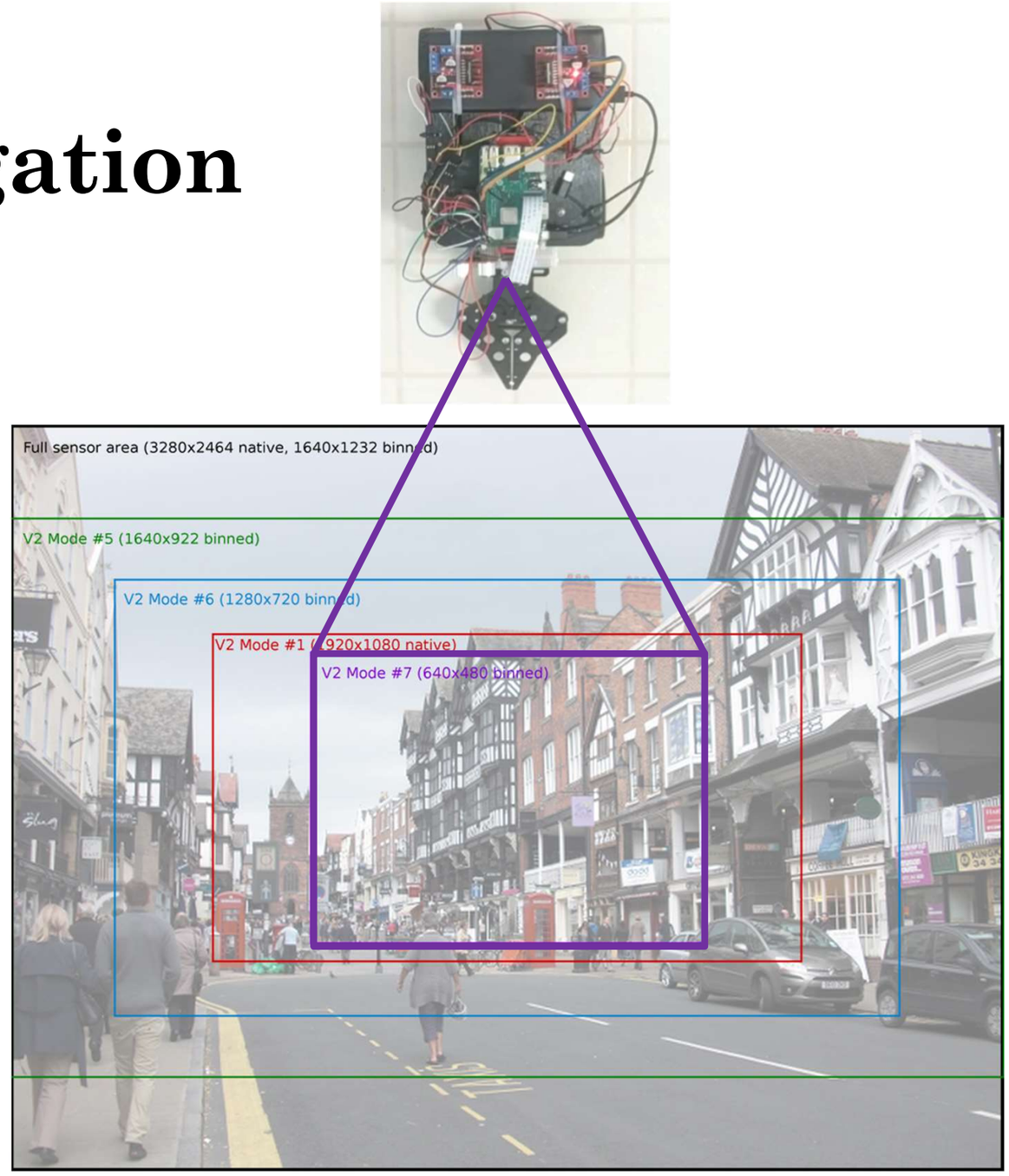
Planning & Navigation

- To navigate, consider the camera FOV

https://picamera.readthedocs.io/en/release-1.12/fov.html

Planning & Navigation

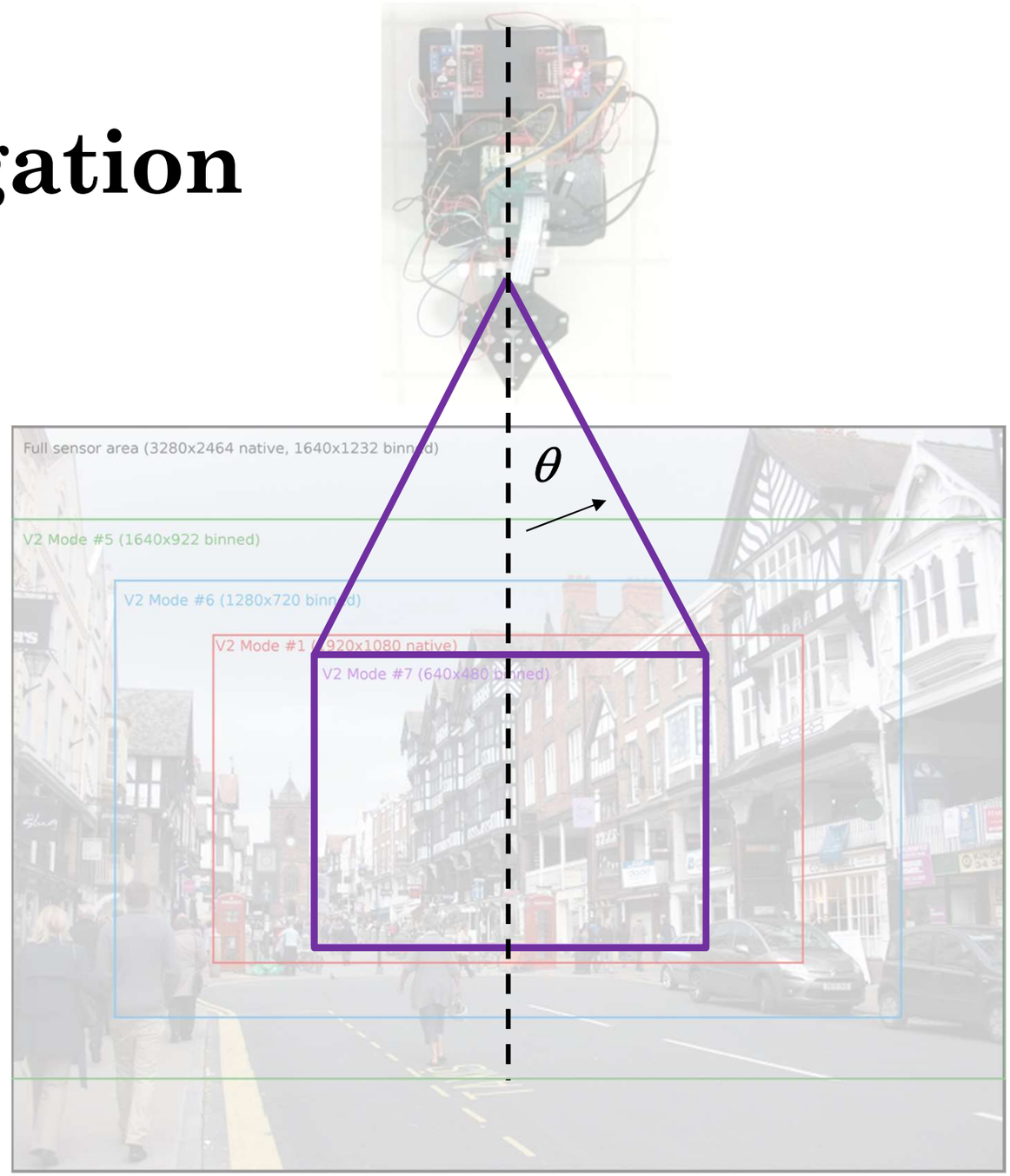
- To navigate, consider the camera FOV
- Horizontal FOV filled up by 12 inch ruler at a 17 inch range



https://picamera.readthedocs.io/en/release-1.12/fov.html

Planning & Navigation

- To navigate, consider the camera FOV
- Horizontal FOV filled up by 12 inch ruler at a 17 inch range
- Thus half angle θ = atan(6/17) = 19.44 deg
- Full angle 2θ = 2*19.44 = 38.88 deg

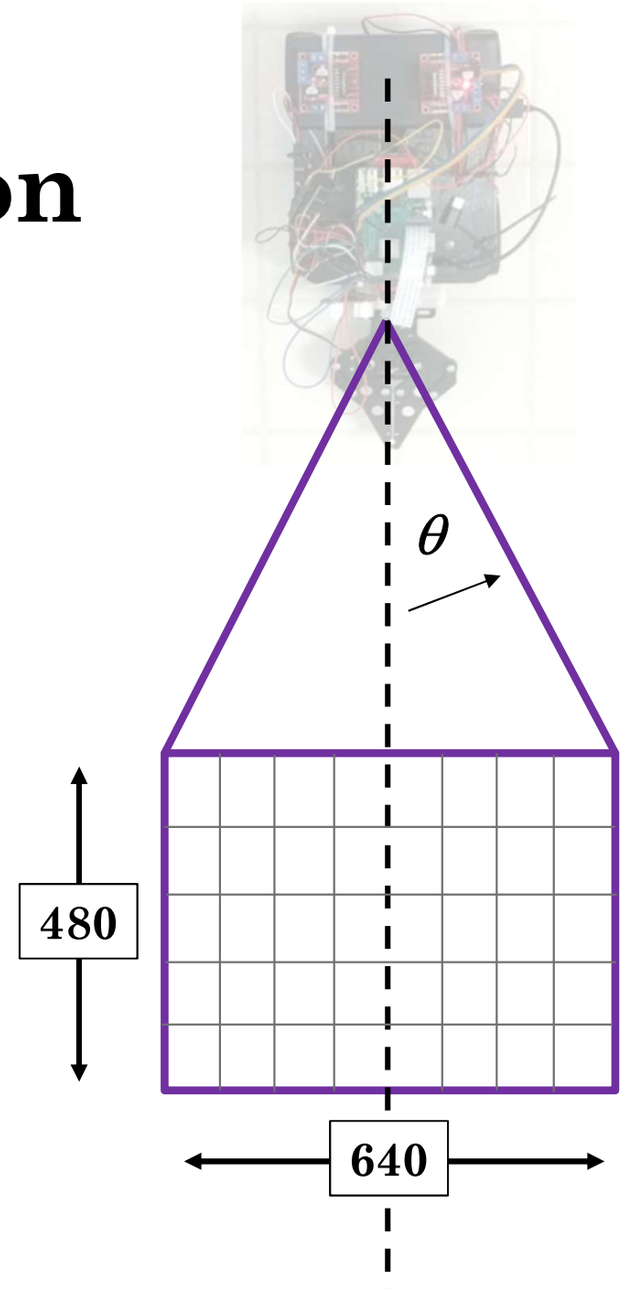


https://picamera.readthedocs.io/en/release-1.12/fov.html

Planning & Navigation

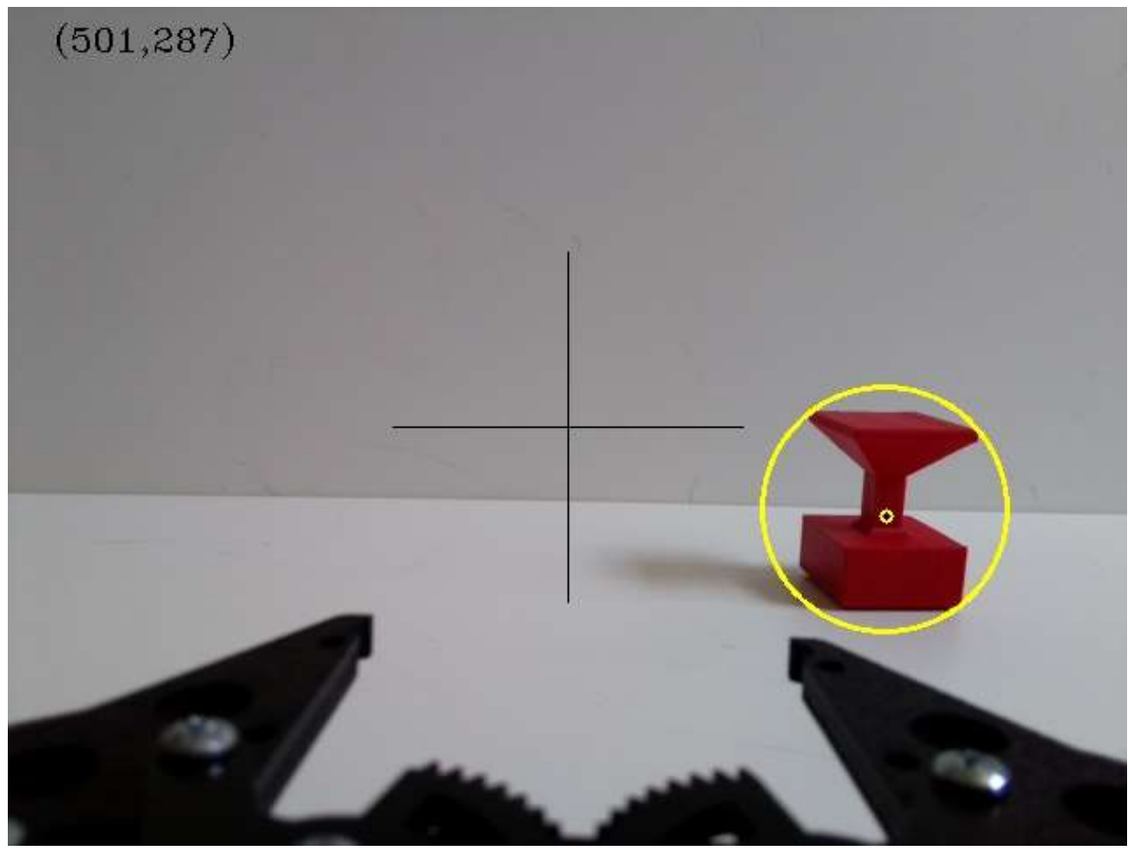
- The 2θ = 38.88 deg FOV is spread out across 640 pixels horizontally

- Therefore **each pixel** contributes 38.88/640 = 0.061 deg



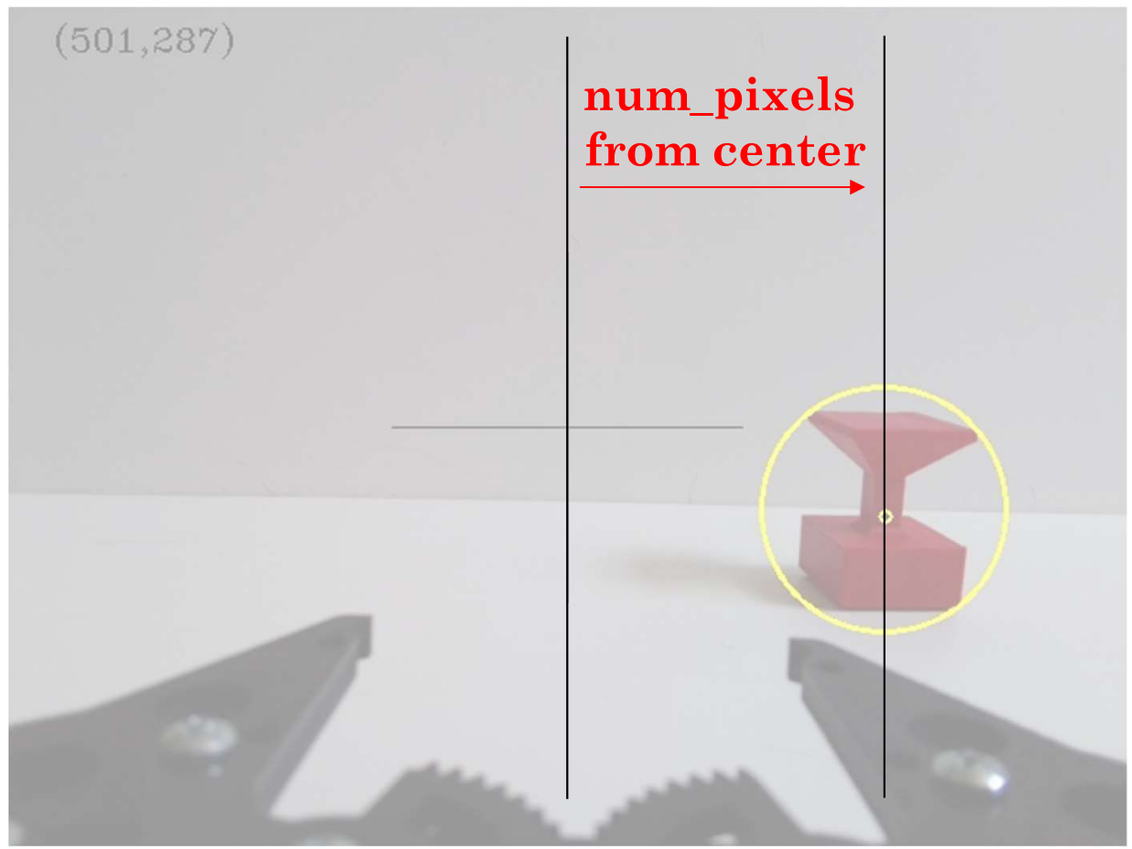
Planning & Navigation

- To align robot with the target contour, rotate by (num_pixels from center to centerline)*0.061



Planning & Navigation

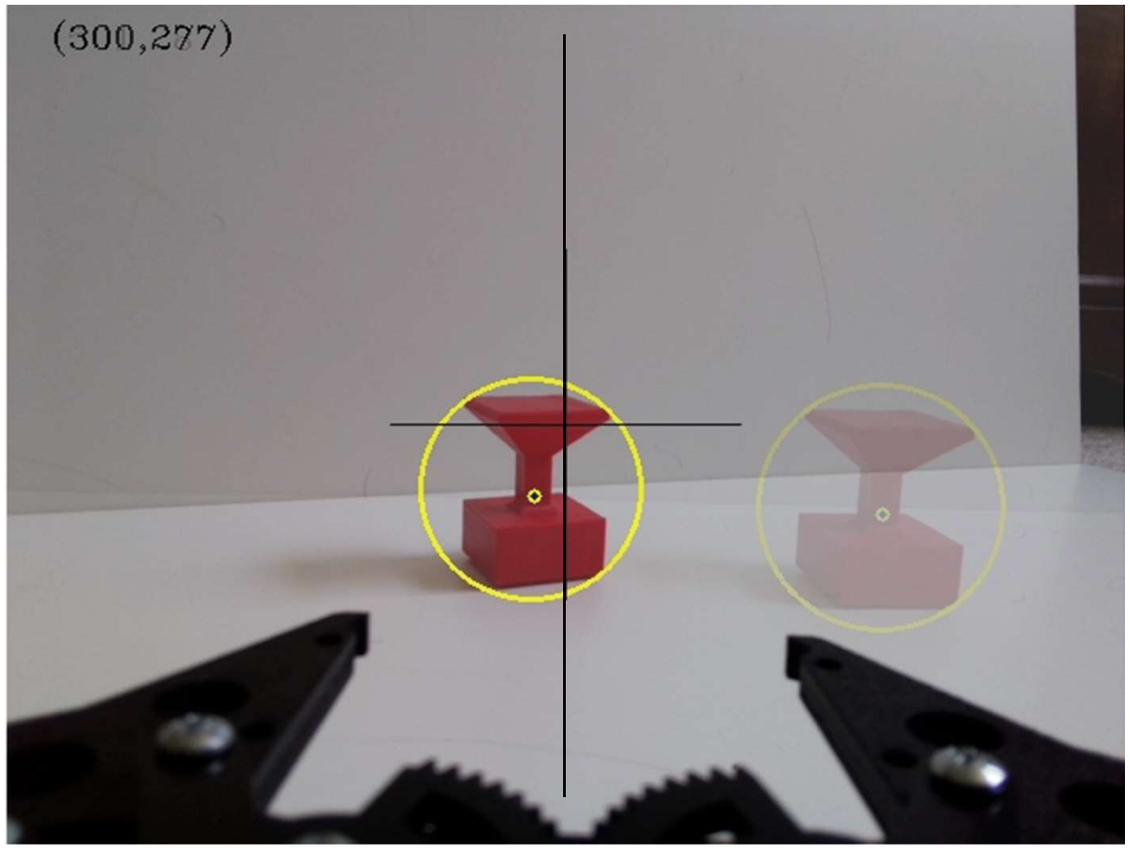
- To align robot with the target contour, rotate by (num_pixels from center to centerline)*0.061
- For example:
- (501-(640/2))*0.061= 11.04 deg



- If contour in RHP, pivotright( )
- If contour in LHP, pivotleft( )

Planning & Navigation

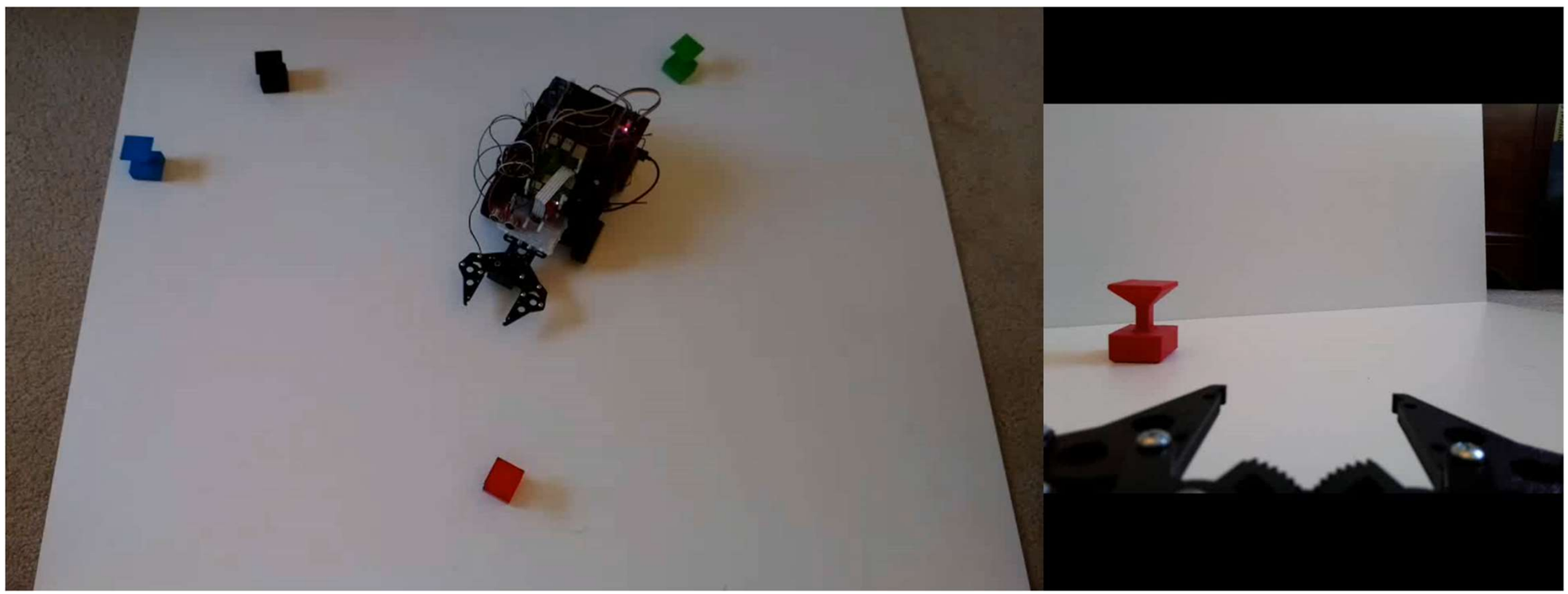
- To align robot with the target contour, rotate by (num_pixels from center to centerline)*0.061
- For example:
- (501-(640/2))*0.061= 11.04 deg



- If contour in RHP, pivotright( )
- If contour in LHP, pivotleft( )

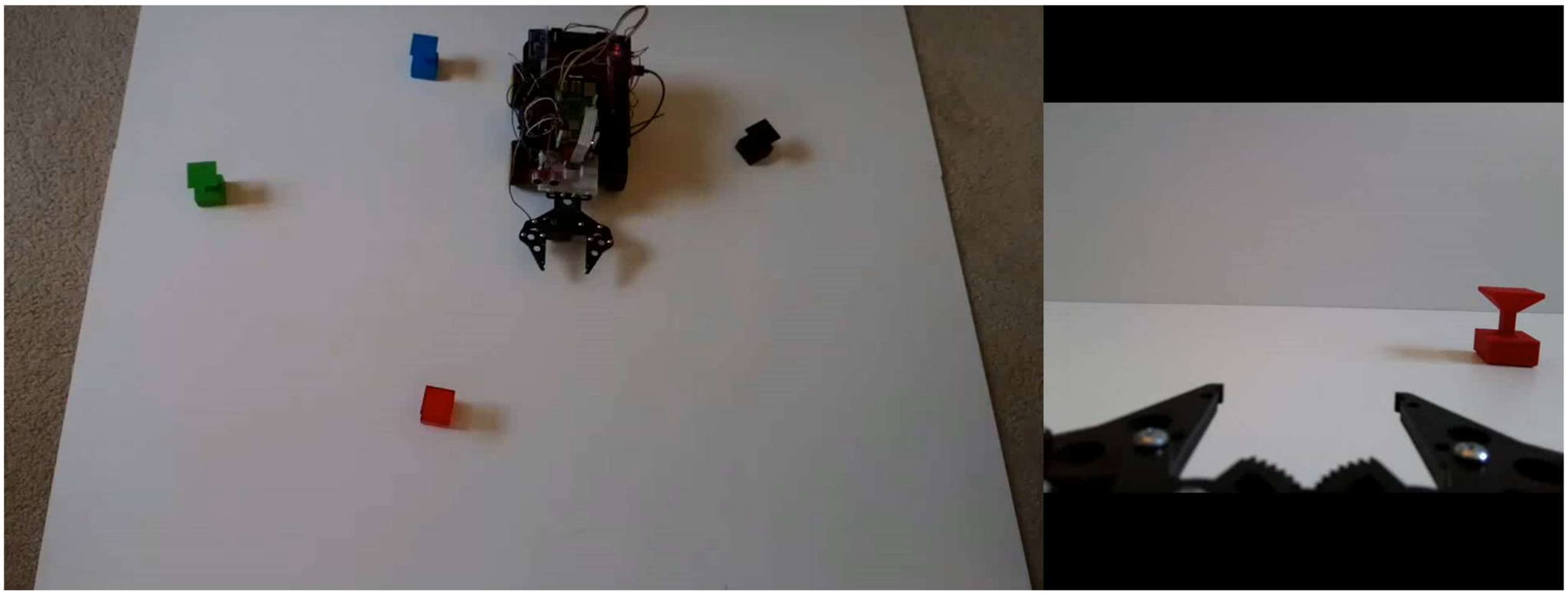
In-Class Exercise

- Create a new Python script: ***trackblock01.py***
- Demonstrate robot can autonomously rotate & track object



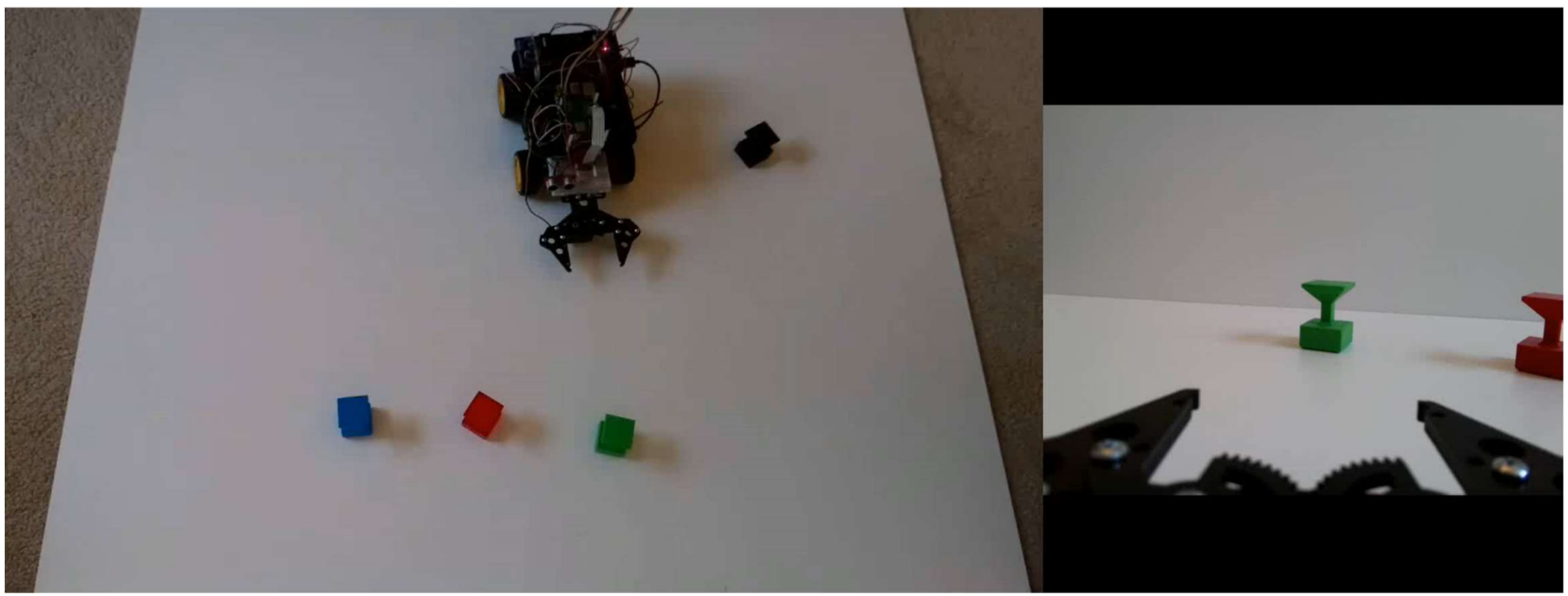
Planning & Navigation

- Update script to enable robot to autonomously retrieve object



In-Class Exercise

- Demonstrate robot autonomously retrieves object in a cluttered environment
- Once retrieved, deliver image via email: **ENPM809TS19@gmail.com**



References

- *Introduction to Autonomous Mobile Robots*, Siegwart
  - Chapter 6
- Picamera: Camera Hardware
  - https://picamera.readthedocs.io/en/release-1.12/fov.html