

USI - MSC AI - ROBOTICS

RoboControl Group

Contents

Objectives

- The objective of our project and system developed

Architecture

- Segmentation of the system and work distribution

Issues

- Simulation adaptation and gimbal issues

Demos

- Video demos on robomaster

The Objective

To control remotely the robomaster by using body gestures.

Architecture

To accomplish our objective, we decided to split the work on 3 different nodes, by implementing the main features of our system

User Tracking

This node controls the gimbal and moves the camera to follow the user. As a result, the camera remains focused on the user, even if the user moves or the RoboMaster rotates.

Gesture Detection

For each frame, this node detects the user's gesture and converts it into a suitable format for processing by the next node.

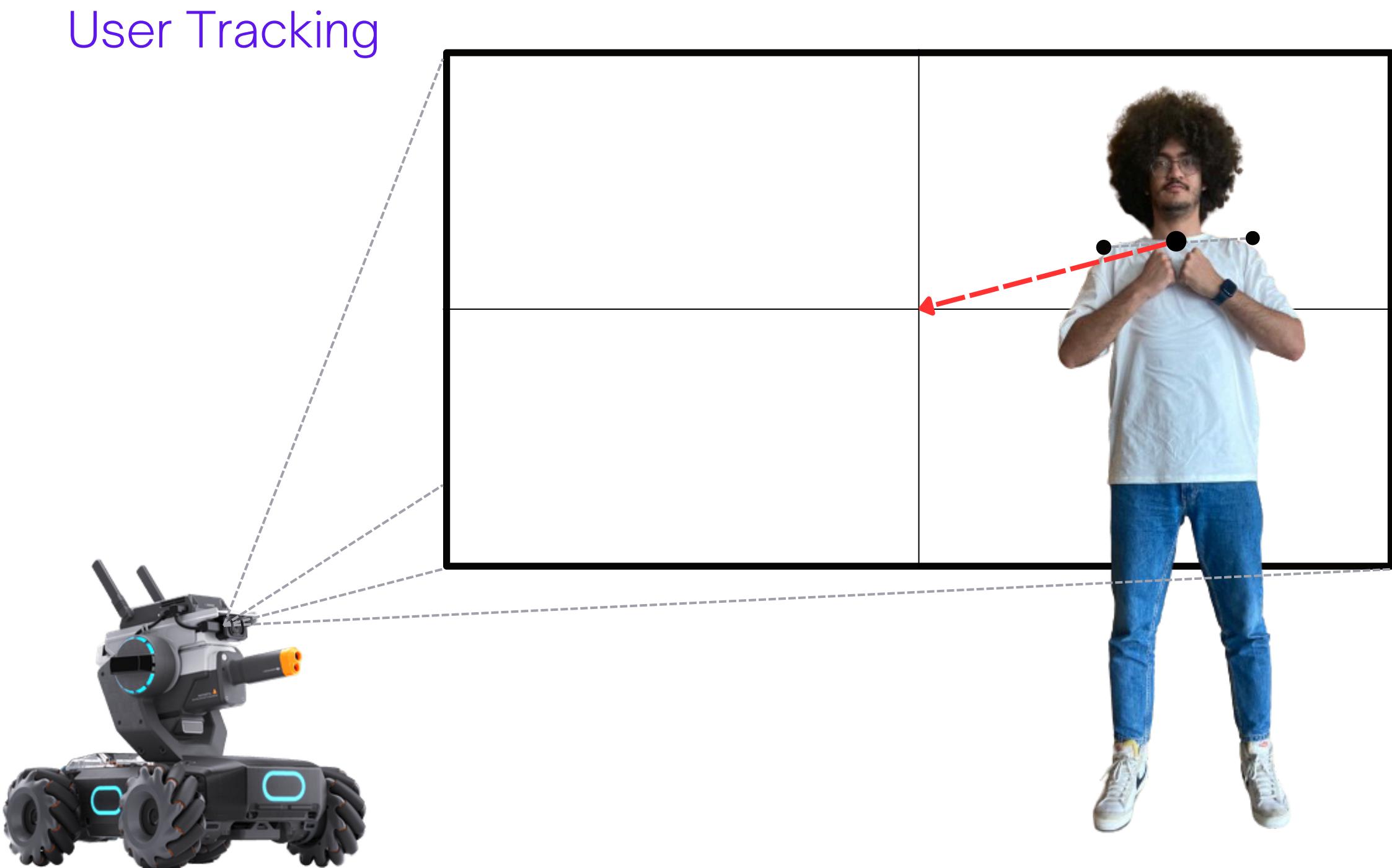
Movement Control

This node receives a message from the previous node and moves the RoboMaster in a specific direction based on the user's gesture.

Architecture

User Tracking

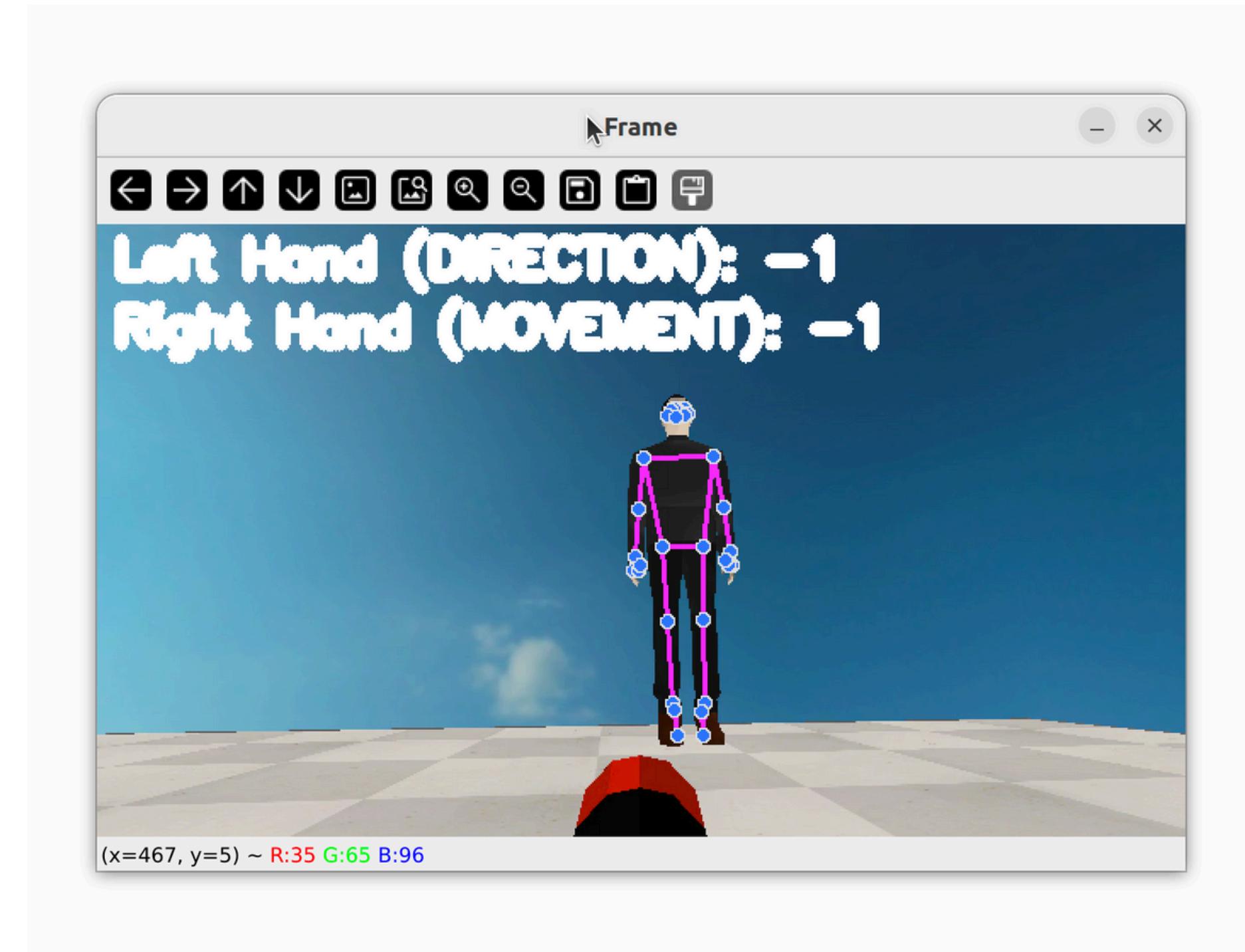
This node controls the gimbal and moves the camera to follow the user. As a result, the camera remains focused on the user, even if the user moves or the RoboMaster rotates.

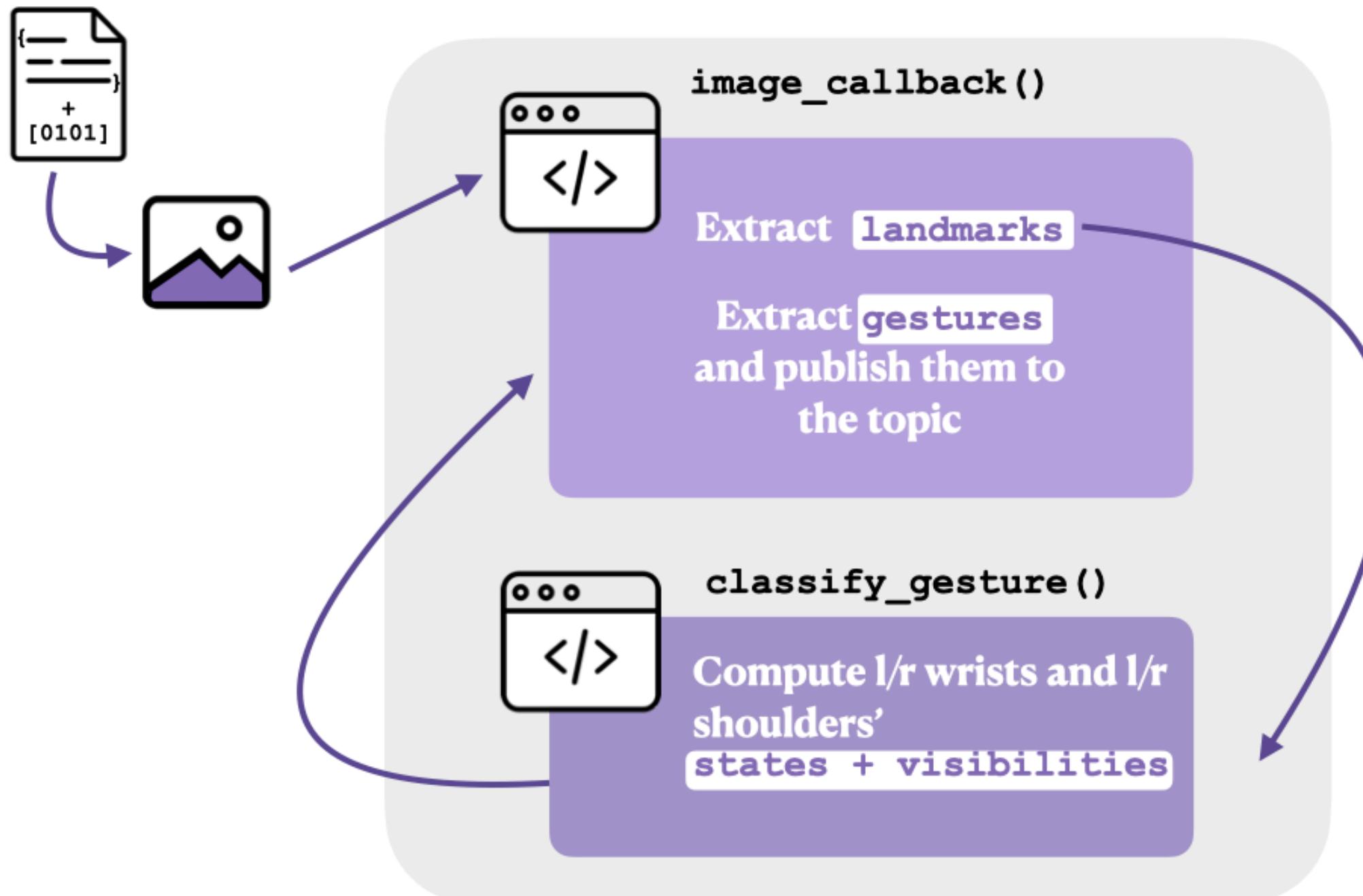


Architecture

Gesture Detection

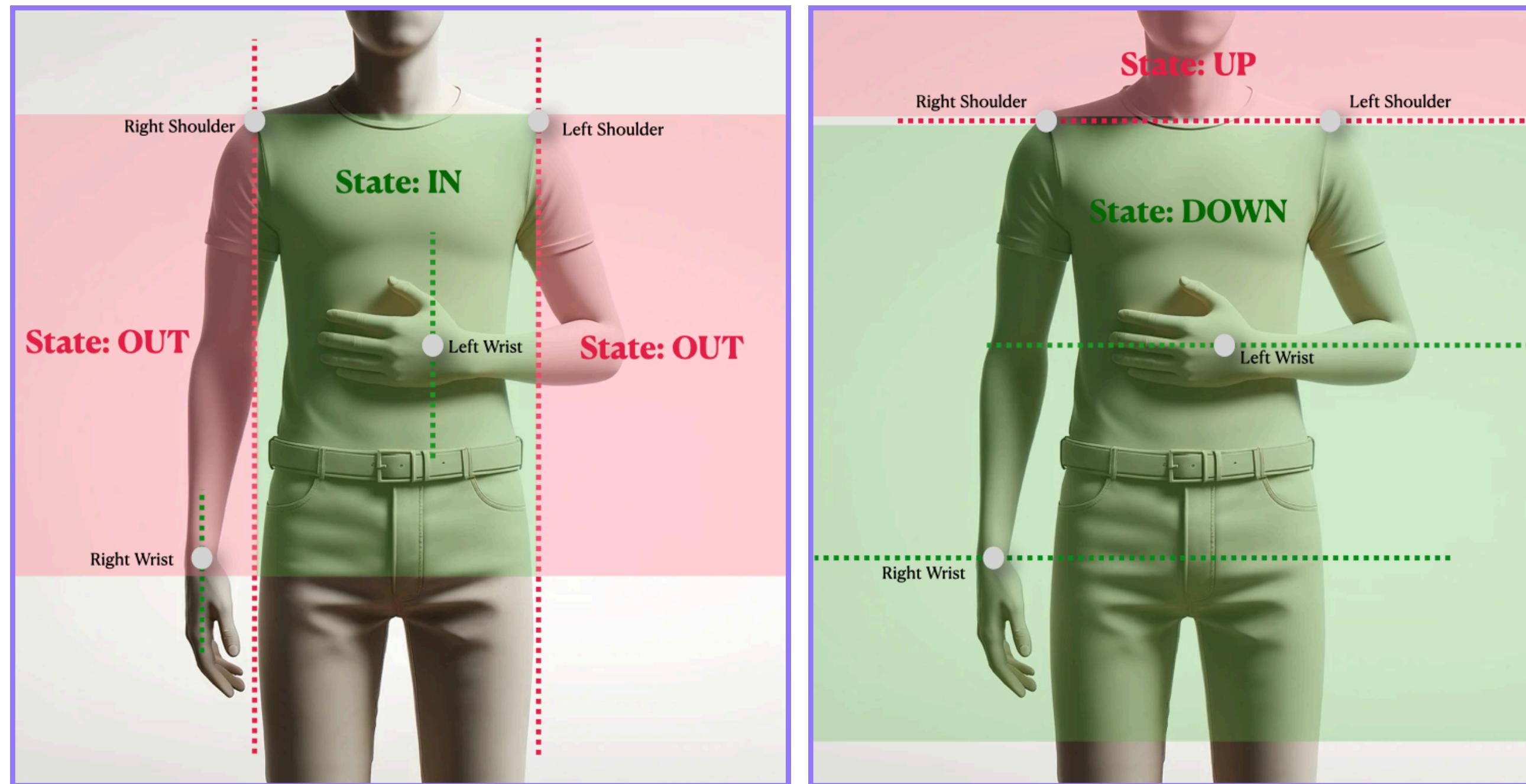
This system takes video input from the ***/camera/image_color*** topic and outputs inferred gestures to the ***/cmd_command*** topic using Google's mediapipe library.





```
#direction  
match (left_state, left_vertical):  
    case ('Out', 'Up'):  
        direction = '1' #'LEFT'  
    case ('Out', 'Down'):  
        direction = '-1' #'RIGHT'  
    case ('In', _):  
        direction = '0' #'STOP'  
  
#movement  
match (right_state, right_vertical):  
    case ('Out', 'Up'):  
        movement = '1' #'FORWARD'  
    case ('Out', 'Down'):  
        movement = '-1' #'BACKWARD'  
    case ('In', _):  
        movement = '0' #'STOP'
```

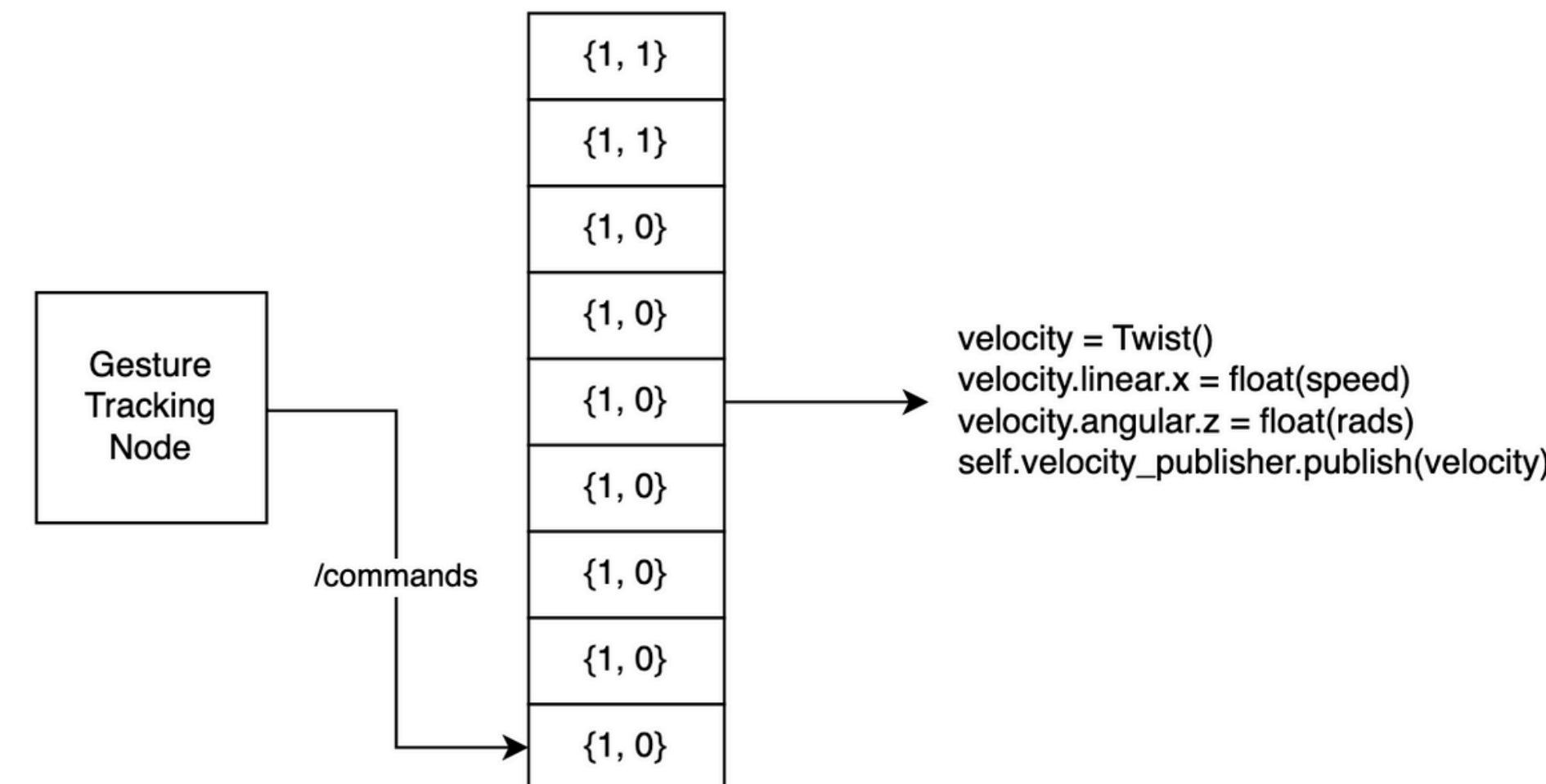
How do we define gestures?



Architecture

Movement Control

msg	Direction	Movement
{1,1}	left	F
{1,0}	left	STOP
{1,-1}	left	B
{0,1}	STOP	F
{0,0}	STOP	STOP
{0,-1}	STOP	B
{-1,1}	right	F
{-1,0}	right	STOP
{-1,-1}	right	B



Issues

During the implementation and integration of this system, we've found the following problems

Simulation Adaptation

Simulating the system proved to be a challenge due to mediapipe issues

Using human models

We tried to use Bob, the human model and also the orange mannequin to perform the gestures.



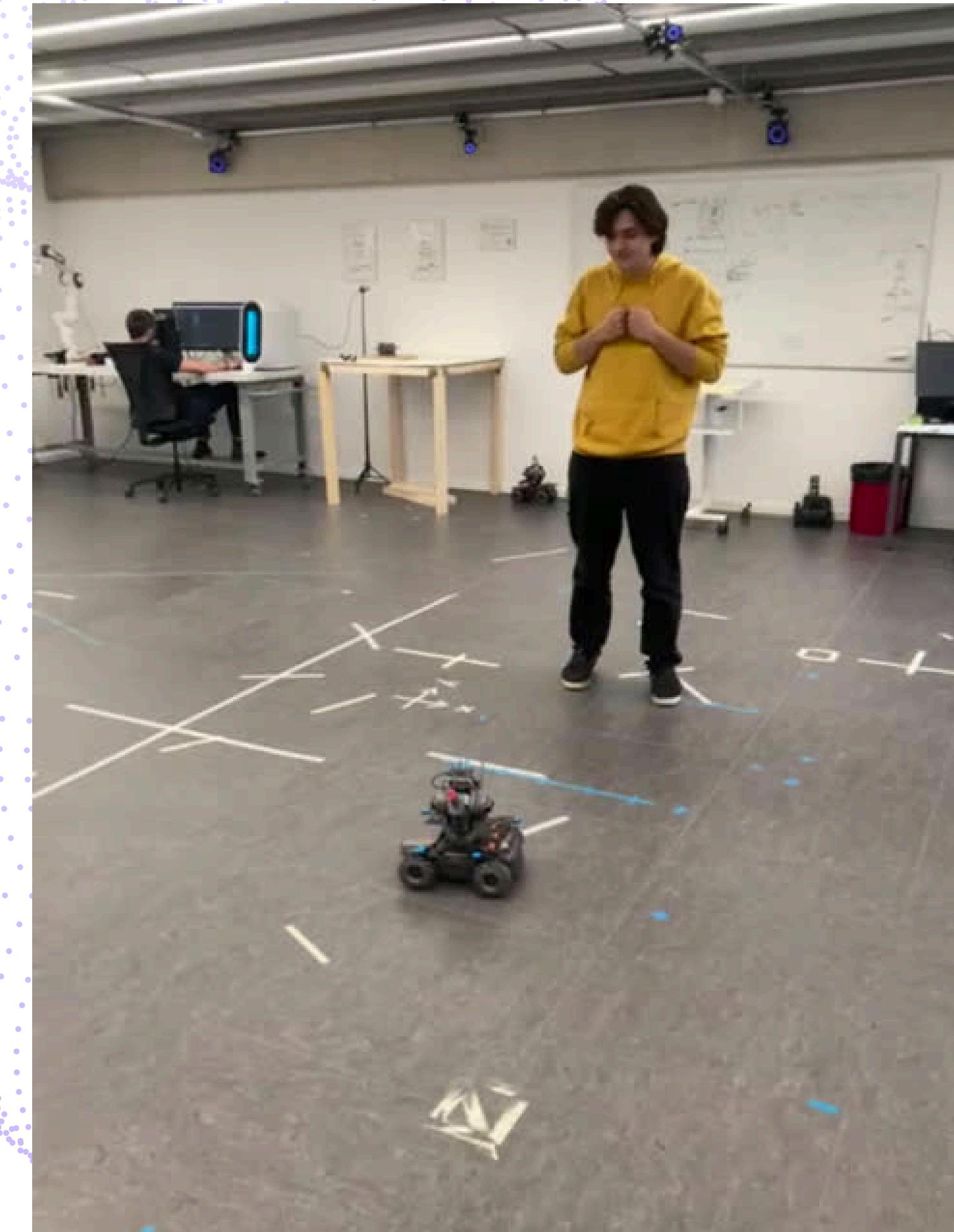
Switch to tables!

After several trials and errors, we decided to use an upside-down table and attached a picture of ourselves to it. This solution worked perfectly.



Gimball Ghost

There is an issue affecting all Robomasters where the gimbal starts moving on its own if the Robomaster is on for more than 15 minutes.



Demo



Thank you

:)

