Marker Follower

A Python-based system for detecting ArUco markers in images or video, and computing robot motion commands to follow a specific marker. This repository includes:

- A reusable marker detection class
- A standalone video processing script
- A ROS node for real-time robot control

Repository Contents

`marker_detector.py`

- `MarkerDetector` class.
- Member methods:
  - `detect()`: Detect ArUco markers in the image frame:
    - Uses OpenCV's ArUco module to:
    - Detect markers in images.
    - Estimate each marker's 3D pose (`rvec`, `tvec`) relative to the camera.
  - `compute_velocity()`: Computes linear and angular velocities based on:
    - Marker distance from the camera.
    - Pixel offset of the marker's center from the image center.
- Parameters:
  - Camera intrinsic parameters: `camera_matrix`, `dist_coeffs`

`run.py`

- A python script to:
  - Load a video file.
  - Detect markers frame-by-frame.
  - Compute desired robot velocities for following marker ID 1.
  - Overlay:
    - Marker bounding boxes.
    - Marker ID text.
    - Velocity vectors and debug info directly on the video frames.
- Draws arrows or curves representing motion commands:
  - Straight arrows for pure forward/backward movement.
  - Curved paths for simultaneous rotation and translation.
- Allows saving the annotated video if desired.
- Inputs:

PROF

- video_path: (e.g.,"/home/kimkt0408/bagfiles/2025-06-30-08-20-43.mp4")

- Parameters:
  - Controller parameters:
    - desired distance
    - k_linear
    - k_angular
    - max_linear_speed
    - max_angular_speed

Example usage:

```
python3 run.py
```

---

marker_follower_ros.py

A ROS wrapper of marker_detector.py

- Subscribed rostopic:
  - Camera images from /usb_cam/image_raw.
  - How to rosrun usb_cam in Clearpath Jackal:

    ```
    rosrun image_view image_view image:=/usb_cam/image_raw
    ```

- Published rostopic:
  - Publish robot velocity commands on /cmd_vel.
  - Publish annotated images for debugging on /marker_follower/annotated_image.

PROF

- Parameters:
  - Controller parameters:
    - desired distance
    - k_linear
    - k_angular
    - max_linear_speed
    - max_angular_speed

Example usage:

```
rosrun marker_follower marker_follower_ros.py
```