Modular Tracking

Leon Oleschko

17.03.2025

Real-time detection, tracking, and control are fundamental to applications in robotics and automation. This project presents a modular, high-speed camera tracking system designed to explore detection, tracking, and control strategies by following a ping pong ball during gameplay. The system combines a Raspberry Pi Camera V2, stepper motors, and a Raspberry Pi 5, achieving real-time tracking at 200 Hz through a modular Python-based software stack.

Introduction

Real-time object tracking is a core challenge in computer vision and robotics. Moving a camera to track a ping pong ball during gameplay creates an interesting platform to explore and evaluate detection, tracking, and control strategies. This is due to its unique combination of steady ballistic flight and rapid, complex bounces at high speeds.

To simplify the exploration of different strategies, the system is designed with modularity and accessibility in mind. Each component is built around high-level concepts, enabling users to experiment without requiring an in-depth engineering background.

1 Methods

The system is designed to track a ping pong ball in real time by moving a camera dynamically. To achieve this, the project is broken down into modular components, each addressing a specific aspect of the tracking process.

Hardware

The hardware components consists of a camera, motors, motor controllers and a processing computer.
For the camera a Raspberry Pi Camera V2 is used, as with the picamera2 software stack it supports a framerate of 200 Hz. A normal 30 Hz USB Webcam was tested to be not sufficient.
To move the camera 2 common stepper motors are mounted perpendicularly. They are driven with field oriented controllers with the SimpleFOC firmware stack. They are connected to the control computer using the Universal Serial Bus (USB).
The chosen computer is a Raspberry Pi 5.

To power the components 9 V 3 A from a USB-C power supply are converted to a 5 V 5 A bus for the computer and another 5 V 3 A bus for the motors. Additionally the startup of the computer and the motors is staggered. This was necessary to achieve stable operation.

To mount the components mechanically, they are designed as subrack modules.

Software

The software architecture is designed to be modular, ensuring flexibility and ease of experimentation. It is split into a camera module, detector module, and motor module, with plans to add a tracker module and controller module in the future. Each module can be independently evaluated, modified, or replaced, allowing users to experiment with different algorithms and configurations without affecting the entire system.

The camera module configures the camera into a high frame rate mode and fetches new frames.

The detector identifies the center of the ping pong ball in the image by comparing each pixel's color $I(x, y)$ to a predefined reference color R in the Hue Saturation Luminance (HSL) color space. Testing showed that HSL outperformed Luminance-AB (LAB). For each pixel, a weight w is computed:

$$w = \exp\left(-\frac{(I(x, y) - R)^2}{\sigma}\right)$$

Where σ is the standard deviation of each channel in a reference image of the ball. This function emphasizes pixels closely matching the reference color while exponentially reducing the influence of outliers. The color channels are then summed to get one weight for each pixel. The location of the ball is determined by finding the position of the maximum weight in the image.
This process is efficiently implemented using numpy, allowing the detector to process an downsampled 80×60 pixel frame in $3.6\,\mathrm{ms}$. This high-speed performance is critical for real-time tracking, ensuring the system can keep up with the fast-moving ball during gameplay.

The tracker evaluates the detector's maximum weight w_{max} to determine if the detection is valid. Using hysteresis, it compares w_{max} to two thresholds $T_{\text{heigh}} > T_{\text{low}}$. If $\omega_{\text{max}} > T_{\text{heigh}}$ it is valid; if $\omega_{\text{max}} < T_{\text{low}}$ it is invalid. For values In between $T_{\text{low}} < \omega_{\text{max}} < T_{\text{heigh}}$ the pervious state is kept.
This hysteresis-based approach ensures robust and smooth tracking, even under challenging conditions.

The motors are moved based on the error, which is the angle between the detected ball position and the center of the camera's field of view. This error is scaled by a constant factor, producing a control signal that drives the motors. This approach is known as a P-regulator (proportional controller), where the motor response is directly proportional to the error. While simple, this method provides effective and responsive control for keeping the ball centered in the frame.

2 Results

In preliminary testing the simple approach works smoothly under controlled lighting conditions.

3 Conclusion and Outlook