Lab 10: Real-Time Object Tracking with Color Filtering

**Introduction**

In this lab, students will learn how to implement real-time object tracking using color filtering techniques. Object tracking is a fundamental concept in computer vision and is used in applications such as surveillance, robotics, and augmented reality. By the end of this lab, students will create a program that tracks objects of a specific color using a webcam feed and highlights the detected object with a bounding box or a circle.

**Relevant Background**

Color filtering is a common technique for detecting objects based on their color. It involves converting an image to the HSV (Hue, Saturation, Value) color space and defining a color range for segmentation. HSV is preferred over RGB for this task as it allows better separation of color and intensity information. After segmentation, contours are used to identify the boundaries of the detected object.

**Steps for the Lab**

**Part 1: Convert Video to HSV and Filter by Color**

1. Capture live video using a webcam.
2. Convert the video frames from BGR to HSV.
3. Define a specific color range to filter (e.g., red, blue, or green).
4. Create a binary mask to isolate the selected color.

**Part 2: Detect Contours and Track the Object**

1. Apply contour detection on the binary mask.
2. Find the largest contour (if any) and compute its bounding box or centroid.
3. Highlight the detected object in the original video feed with a bounding box or circle.

**Part 3: Implement Object Size Tracking**

1. Measure the size of the detected object.
2. Display the size in the video feed.
3. Optionally, use the size to control an action (e.g., display a warning if the object size exceeds a threshold).

**Grading**

* **10 Points**: Add your name and date to the beginning of your code.
* **20 Points**: Use appropriate comments throughout your code.
* **40 Points**: Working real-time color filtering and object tracking.
* **30 Points**: Correct implementation of bounding box and size display.

**Example Outputs**

1. **Binary Mask**: A black-and-white image showing the filtered color.
2. **Tracked Object**: A bounding box or circle drawn on the object in the live feed.
3. **Size Feedback**: Text displaying the object's size overlayed on the video.

Example 1

A person holding a piece of paper with a green rectangle

Description automatically generated

Example 2  
  
A screenshot of a computer

Description automatically generated