CU Boulder
Mechanical Engineering
Fall 2023

MCEN 5115 – Mechatronics and Robotics - I

Graduate Project Design Report

Allister James Sequeira

12/18/2023

Object Tracking Camera

Introduction: The Object Tracking Camera used an Arduino, two servo motors and a Pixy Camera for automated tracking of objects within a given environment.

Hardware Configuration:

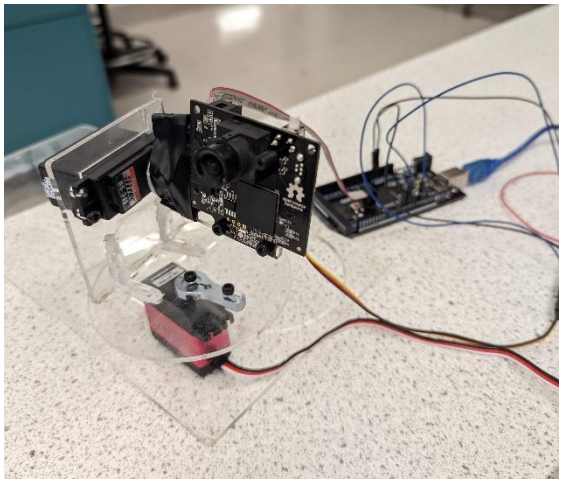
Pixy Camera: The Pixy Camera is used for its color-based object recognition. Trained color signatures enable the camera to identify and track specific objects.

Servo Motors: Two servo motors control the pan and tilt movements of the Pixy Camera. The range of each of the servos is 270 degrees.

Arduino Board: The Arduino board acts as the brain of the system, processing information from the Pixy Camera and controlling the servo motors.

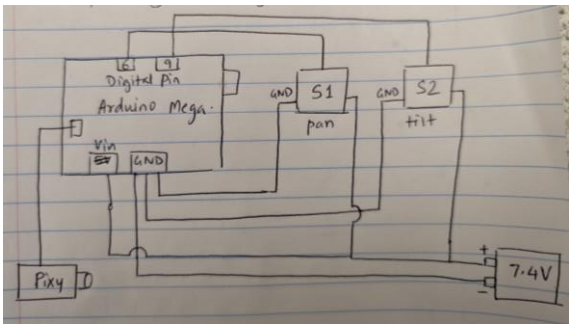
Mechanical Frame: A acrylic sheet frame holds the Pixy Camera and servo motors.

Power Supply: A 7.4V LiPo battery powers the entire system.



System Operation:

Pixy Camera detects an object, a trained pink color in this case, based on their color signatures. The pan and tilt servo motors adjust the camera's orientation based on the detected object's position. The x coordinate of the detected object is used to adjust the pan motion and the y coordinate is used for the tilt motion. Real-time adjustments maintain the object within the camera's field of view. Arduino processes Pixy Camera data and sends commands to servo motors. The code ensures smooth and responsive tracking of objects. The power supply to the servo motors is from a 7.V LiPo battery. The grounds of the servo and the Arduino need to be connected to avoid erratic motion of the servo. Circuit diagram is as per following sketch.



Applications:

Surveillance and Security: The Object Tracking Camera can be used for automated surveillance in sensitive areas. Continuous monitoring and tracking improve security measures.

Automated Photography and Videography: Best cinematic shots use object detected camera which seamlessly tracks and captures dynamic scenes without manual camera adjustments. This is good for sports events, wildlife photography, and high-speed cinematic shots.

Other applications can be in Human-Computer Interaction, Education and IoT.

Conclusion:

The Object Tracking Camera is an efficient solution for real-time object tracking. The integration of color-based recognition with a pan-and-tilt mechanism offers flexibility in applications ranging from surveillance to creative photography.

Arduino Code:

```
#include <Servo.h>
#include <Pixy.h>

Pixy pixy;
int x = -10;
int y = -10;
int width = -10;
int height =-10;

int x2deg = 90;
int y2deg = 90;
int xservo_pin = 6;
int yservo_pin = 9;

Servo myservoX;  // create servo object to control a servo
Servo myservoY;  // twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  //Serial.println("HERE");
  SPI.begin();
  Serial.begin(115200);
  myservoX.write(90);
  myservoY.write(90);
  myservoX.attach(xservo_pin);  // attaches the servo on pin 9 to the servo
object
  myservoY.attach(yservo_pin);
  pixy.setBrightness(100);
  pixy.init();
  // pixy.changeProg("color_connected_components");
}
int blocks;
int i = 0;
int count = 0;

void loop()
{
  blocks = pixy.getBlocks();

 if (blocks)
  {
    x = pixy.blocks[0].x;         //The x location of the center of the
detected object (0 to 316)
    Serial.print("OBJ X POS: ");
    Serial.println(x);
```

```
    if (x < 148 && x > 0)
    {
      if (x2deg >= 180)
      {
        x2deg = 90;
        delay(500);
      }
      else
      {
        x2deg = x2deg + 1;
        myservoX.write(x2deg);
      }
    }
    else if (x > 168 && x < 316)
    {
      if (x2deg <= 0)
      {
        x2deg = 90;
        delay(500);
      }
      else
      {
        x2deg = x2deg - 1;
        myservoX.write(x2deg);
      }
    }

    y = pixy.blocks[0].y;//The y location of the center of the detected object
(0 to 200)
    Serial.print("OBJ Y POS: ");
    Serial.println(y);

    if (y < 90 && y > 0)
    {
      if (y2deg >= 180)
      {
        y2deg = 90;
        delay(500);
      }
      else
      {
        y2deg = y2deg + 1;
        myservoY.write(y2deg);
      }
    }
    else if (y > 110 && y < 200)
    {
      if (y2deg <= 0)
```

```
      {
        y2deg = 90;
        delay(500);
      }
      else
      {
        y2deg = y2deg - 1;
        myservoY.write(y2deg);
      }
    }
    delay(10);
  }
}
```