

UNIVERSIDAD POLITÉCNICA SALESIANA

NAMES: JOSÉ ILLESCAS, OSCAR MORA

GROUP: 1

DATE: 12-AGOSTO-2019

VISIÓN ARTIFICIAL POR COMPUTADORA

OBJECTIVES

- Learn to use the different commands and libraries of the OpenCV language.
- Recognize objects and colors using artificial vision.

PURPOSE

With this project we want to incorporate the different functionalities that our robotic prototype will have that will help people with brain disabilities.

DEVELOPING

The following project was carried out using a library free of artificial vision called OpenCV, with which this library has several applications such as motion detection and recognition of objects and colors.

These applications were made in a robotic prototype, which will serve to help children with brain disabilities, what was done in this project was an incorporation of the two applications that are color and movement detection.

This project incorporates different modules that are coding using OpenCV, coding in the Arduino IDE which will help us in the operation and interaction with the robotic prototype. Within the operation we also have the functionality to speak identifying the different colors that are shown.

OPENCV CODE

```
import cv2
import numpy as np
import serial
import time

def dibujar(mask, color):
    contornos,_ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    #cv2.drawContours(frame, contornos, -1, (255,0,0), 3)
```

```

#cv2.imshow('maskAzul', mask)

# este bucle calcula el área del objeto seleccionado para graficarlo
for c in contornos:
    area = cv2.contourArea(c)
    if area > 3000:
        #Calculamos el centro del objeto seleccionado
        M = cv2.moments(c)
        #controlamos la división para q no sea 0
        if (M["m00"]==0): M["m00"]=1
        x = int (M["m10"]/M["m00"])
        y = int (M["m01"]/M["m00"])
        cv2.circle(frame, (x,y), 7, (0,255,0), -1)
        #Fuente del texto para el valor del centro del objeto
        font = cv2.FONT_HERSHEY_SIMPLEX
        #colocación del valor del centro
        #cv2.putText(frame, '{}{}'.format(x,y),(x+10,y), font,
0.75,(0,255,0),1,cv2.LINE_AA)
        print(x)
        #Enviamos el valor de servo a arduino para el giro de la cabeza
        #arduino = serial.Serial('COM3', 9600)
        #time.sleep(2)
        print("Connection to arduino...")
        face_cascade =
cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
        print ('X :'+str(x))
        xx = int(x+(x+1))/2
        yy = int(y+(y+1))/2
        center = (xx,yy)
        data = 'X{0:.0f}Y{1:.0f}Z'.format(xx, yy)
        print ("output = '" +data+ "'")
        arduino = serial.Serial('COM3', 9600)
        arduino.write(str.encode(data))

        #arduino.write(x)
        #print(arduino.readline())
        #leer=arduino.readline()
        #print(leer)
        #time.sleep(1)
        #arduino.close()
        #Eliminar los picos del contorno y suaviza el contorno al objeto
        nuevoContorno = cv2.convexHull(c)
        # dibuja solo los contornos superiores a esa area
        cv2.drawContours(frame, [nuevoContorno], 0, color, 3)

captura = cv2.VideoCapture(0);
#azulBajo = np.array([100,100,20], np.uint8)
#azulAlto = np.array([125,255,255], np.uint8)

```

```

amarilloBajo = np.array([15,100,20], np.uint8)
amarilloAlto = np.array([45,255,255], np.uint8)

#rojoBajo1 = np.array([0,100,20], np.uint8)
#rojoAlto1 = np.array([5,255,255], np.uint8)

#rojoBajo2 = np.array([175,100,20], np.uint8)
#rojoAlto2 = np.array([179,255,255], np.uint8)

font = cv2.FONT_HERSHEY_SIMPLEX

while True:
    ret, frame = captura.read()
    if ret == True:
        frameHSV = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
        #maskAzul = cv2.inRange(frameHSV, azulBajo, azulAlto)
        maskAmarillo = cv2.inRange(frameHSV, amarilloBajo, amarilloAlto)
        #maskRojo1 = cv2.inRange(frameHSV, rojoBajo1, rojoAlto1)
        #maskRojo2 = cv2.inRange(frameHSV, rojoBajo2, rojoAlto2)
        #maskRojo = cv2.add(maskRojo1, maskRojo2)
        #dibujar(maskAzul,(255,0,0))
        dibujar(maskAmarillo,(0,255,255))
        #dibujar(maskRojo,(0,0,255))
        cv2.imshow('Ventana Principal', frame)
        if cv2.waitKey(1) & 0xFF == ord('s'):
            break
    captura.release()
cv2.destroyAllWindows()

```

ARDUINO CODE

```

//librerias necesarias
#include <SD.h>
#include <SPI.h>
#include <TMRpcm.h> // libreria que permite reproducir sonido en Arduino.
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <SoftwareSerial.h>
SoftwareSerial ModBluetooth(0, 1); // RX | TX

Adafruit_PWMServoDriver servos = Adafruit_PWMServoDriver(0x40);
unsigned int pos0=172; // ancho de pulso en cuentas para pocicion 0°
unsigned int pos180=565; // ancho de pulso en cuentas para la pocicion 180°

TMRpcm Audio; //Se crea un objeto de la librería TMRpcm llamado Audio.

```

```
#define pinSD 10 //Define el pin para seleccionar la tarjeta SD en el modulo es CS
```

```
void setup() {  
  servos.begin();  
  servos.setPWMPFreq(60); //Frecuencia PWM de 60Hz o T=16,66ms  
  Audio.speakerPin = 9; //Define el pin en el que está conectada un parlante.  
  Serial.begin(9600); // Inicia la comunicacion Serial.  
  ModBluetooth.begin(9600);  
    //Serial.begin(9600);  
  ModBluetooth.println("MODULO CONECTADO");  
  ModBluetooth.print("#");  
  
  if (!SD.begin(pinSD)) { //Verifica la coneccion de la tarjeta sd:  
    Serial.println("Fallo en la tarjeta SD"); //Aviso de que algo no anda bien  
    return; //No hacer nada si no se pudo leer la tarjeta  
  }  
}
```

```
void setServo(uint8_t n_servo, int angulo) {  
  int duty;  
  duty=map(angulo,0,180,pos0, pos180);  
  servos.setPWM(n_servo, 0, duty);  
}
```

```
void loop() {  
  setServo(0,90);  
  setServo(1,83);  
  setServo(2,100);  
  setServo(3,105);  
  setServo(4,90);  
  setServo(5,90);  
  setServo(6,110);  
  setServo(7,80);  
  setServo(8,50); //Hombro izquierdo  
  setServo(9,90); //brazo drerc  
  setServo(11,20); //ojos  
  setServo(12,80); //honbro nudo Izquierdo  
  setServo(13,80); //honbro nudo Izquierdo
```

```
if(Serial.available() > 0){  
  
  char dato = Serial.read();  
  
  if (dato == 'a'){  
    Serial.println("Recivi Amarillo");  
    Serial.println(dato);
```

```

        Audio.setVolume(5);
        Audio.play("amarillo.wav");
    }

    if (dato == 'b'){
        Serial.println("azul Azul");
        Serial.println(dato);
        Audio.setVolume(5);
        Audio.play("azul.wav");
    }

    if (dato == 'c'){
        Serial.println("Recivi Rojo");
        Serial.println(dato);
        Audio.setVolume(5);
        Audio.play("rojo.wav");
    }
}

//Movimiento Brazos
    if (ModBluetooth.available()) {
        char VarChar;
        VarChar = ModBluetooth.read();
        if(VarChar == '1') {
            Serial.println("saludo");
            Audio.setVolume(5);
            Audio.play("saludo.wav");
        }
    }
} // Fin del loop

```

VIDEO LINK

https://www.youtube.com/watch?v=vsw9rKHGz9Q&feature=youtu.be&fbclid=IwAR1hjmDmKaULjPmkqLfZOz7fV-xzNTjQsiVSwXWbrgZoGWEH8Ap_M0CVjgg

CONCLUSIONS

- With this project we managed to learn how to use the different libraries that we have of C ++ in order to create a project to help people with disabilities.
- It was possible to learn more about the concept of Artificial vision.

BIBLIOGRAPHY

<https://unpocodejava.com/2013/10/09/que-es-opencv/>

<https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>