UNIVERSIDAD DEL BÍO-BÍO

08 Practicando – Seguimiento pelota roja

Tutorial Procesamiento de Imagen con webcam

Estudiantes Practicantes:
Javiera Gutiérrez
Javiera Henríquez
Teresa Vidal

Profesor:
Luis Vera

Laboratorio CIMUBB

2023-1

Seguimiento pelota roja

Para este proyecto utilizaremos nuestra webcam y haremos que rastree y marque en pantalla una pelota roja.

Creamos otro archivo .py y le importamos las siguientes librerías, en este caso no utilizaremos Tkinter para la creación de la ventana:

05 Seguimiento pelota roja.py > ...

```
1   from collections import deque
2   from imutils.video import VideoStream
3   import numpy as np
4   import argparse
5   import cv2
6   import imutils
7   import time
8
```

Luego, escribimos las siguientes líneas para inicializar nuestra aplicación:

```
10  ap = argparse.ArgumentParser()
11  ap.add_argument("-v", "--video",
12      help="path to the (optional) video file")
13  ap.add_argument("-b", "--buffer", type=int, default=64,
14      help="max buffer size")
15  args = vars(ap.parse_args())
```

Definimos los valores **Low** y **Upper** en los que estará nuestro color Rojo, usando el formato **BGR.**

```
17  # Se definen los valores Low y Upper del color rojo
18
19  redLower = (161, 155, 84)
20  redUpper = (179, 255, 255)
21  pts = deque(maxlen=args["buffer"])
```

Se inicia la captura de video utilizando nuestra webcam y se crea una **mask** para nuestra imagen.

```
if not args.get("video", False):
    vs = VideoStream(src=0).start()
else:
    vs = cv2.VideoCapture(args["video"])
# Tiempo para que la cámara cargue
time.sleep(2.0)

while True:

    frame = vs.read()
    frame = frame[1] if args.get("video", False) else frame

    if frame is None:
        break
    frame = imutils.resize(frame, width=600)
    blurred = cv2.GaussianBlur(frame, (11, 11), 0)
    hsv = cv2.cvtColor(blurred, cv2.COLOR_BGR2HSV)

    mask = cv2.inRange(hsv, redLower, redUpper)
    mask = cv2.erode(mask, None, iterations=2)
    mask = cv2.dilate(mask, None, iterations=2)
```

Con el siguiente código se buscará el punto central de nuestra pelota y se le creará un contorno. Si se detecta uno o varios contornos, el programa analizará cual es el más grande y guarda el valor, también busca cual es el contorno más pequeño. Calcula el Radio de la pelota y su centro.

Sí el radio cumple con un mínimo dado, crea un círculo que va marcando la pelota a medida que se mueve.

Se va actualizando la cola con los puntos obtenidos a medida que se mueve la pelota.

```
    # Encuentra el punto central de la pelota
    # (x, y)
    cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
    cnts = imutils.grab_contours(cnts)
    center = None
    # Se procede solo si encuentra un contorno
    if len(cnts) > 0:
        # Encuentra el contorno más grande y lo usa
        # Crea el contorno menor posible
        c = max(cnts, key=cv2.contourArea)
        ((x, y), radius) = cv2.minEnclosingCircle(c)
        M = cv2.moments(c)
        center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"]))
        # Procesos solo si el radio cumple con el valor mínimo
        if radius > 10:
            # Dibuja el círculo
            # Actualiza la lista de puntos rastreados
            cv2.circle(frame, (int(x), int(y)), int(radius),
                (0, 255, 255), 2)
            cv2.circle(frame, center, 5, (0, 0, 255), -1)
    # Actualiza la cola de puntos
    pts.appendleft(center)
```

Se hace un loop sobre los puntos rastreados y se dibuja una línea siguiendo el recorrido de nuestra pelota.

Se le añade un **if** para cerrar el programa usando la tecla ‘q’.

```
        # Hace loop sobre el conjunto de puntos rastreados
    for i in range(1, len(pts)):
        # Si no hay puntos ignora
        if pts[i - 1] is None or pts[i] is None:
            continue
        # Calcula el grosor de la línea y la dibuja
        thickness = int(np.sqrt(args["buffer"] / float(i + 1)) * 2.5)
        cv2.line(frame, pts[i - 1], pts[i], (0, 0, 255), thickness)
    # muestra la ventana en la pantalla
    cv2.imshow("Seguimiento pelota roja", frame)
    key = cv2.waitKey(1) & 0xFF
    # Cierra el programa al apretar la tecla 'q'
    if key == ord("q"):
        break
if not args.get("video", False):
    vs.stop()
else:
    vs.release()

#Para cerrar este programa hay que hacer clic en la tecla 'q' de nuestro teclado.
```