

# Seguimiento de personas usando CV

Santamarina Gaspar y Afur Matías

29 de julio de 2019

# 1. Introducción

El objetivo de este trabajo práctico es desarrollar un script en Python que permita realizar el seguimiento de múltiples personas, en tiempo real, usando la librería OpenCV. El script fue desarrollado con las versiones 3.0 de Python y 4.1.0 de OpenCV.

## 2. Desarrollo

El mecanismo de seguimiento consta de dos partes; en primer lugar un detector de Hog, previamente entrenado para detectar personas; el detector devuelve un conjunto de bounding boxes dentro de las cuales se encuentran contenidas las personas detectadas. En segundo lugar, luego de realizar la detección, se procede a crear un tracker para cada persona detectada, encargado de trackearla en los sucesivos cuadros.

### 2.1. Detector de personas

La detección de personas se basa en los clasificadores de objetos. Estos clasificadores son entrenados para reconocer patrones en las imágenes a partir de las características de las mismas. OpenCV cuenta con la implementación de uno de estos algoritmos clasificadores, el algoritmo HOG (Histogram of Oriented Gradients), que ya viene entrenado para la detección de personas.

A continuación se presentan las líneas de código para la configuración del detector:

```
hog = cv2.HOGDescriptor()  
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
```

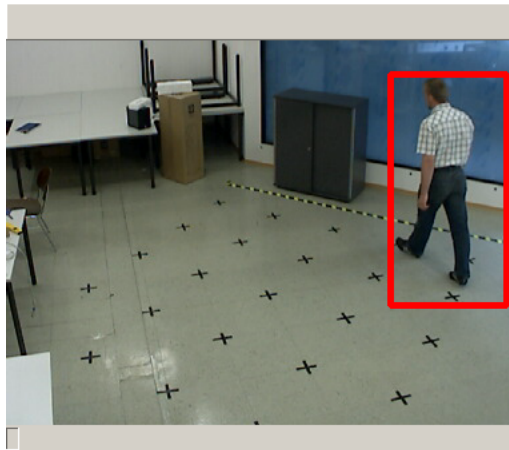


Figura 1: Resultado del detector HOG.

### 2.2. Tracker

OpenCV cuenta con múltiples implementaciones del algoritmo de seguimiento: BOOSTING, MIL, KCF, TLD, MEDIANFLOW, GOTURN, MOSSE and CSRT. Luego de experimentar con las distintas implementaciones, se optó por el tracker KCF, que presentó el mejor rendimiento a lo largo de las pruebas.

Para utilizar el tracker, es necesario inicializarlo con el frame y la bounding box donde se encuentra el objeto a trackear. Para mejorar el rendimiento del mismo, se realiza previamente una substracción del fondo, que consiste en generar una máscara con las zonas donde se encuentran los pixeles en movimiento a lo largo de los frames previos. El código utilizado para dicho procedimiento se presenta a continuación:

```
backSub = cv2.createBackgroundSubtractorKNN()  
mask = backSub.apply(frame)
```

```

tracker = cv2.TrackerKCF_create()
ok = tracker.init(mask, (detected_x, detected_y, detected_w, detected_h))

```

### 2.3. Seguimiento de múltiples personas

Dado que se desean trackear múltiples personas, es necesario contar con un mecanismo que permita diferenciar entre las nuevas detecciones del detector HOG y las que ya están siendo trackeadas.

Para ello se calcula, en cada frame, el area de solapamiento entre cada una de las detecciones del detector y de los trackers actuales. Si dicha area es mayor a un threshold preseteado, se supone que la detección se corresponde con el tracker; si no se encuentra ningún tracker que se corresponda con la detección, se procede a crear un nuevo tracker tomando como bounding box de inicialización la bounding box devuelta por el detector.

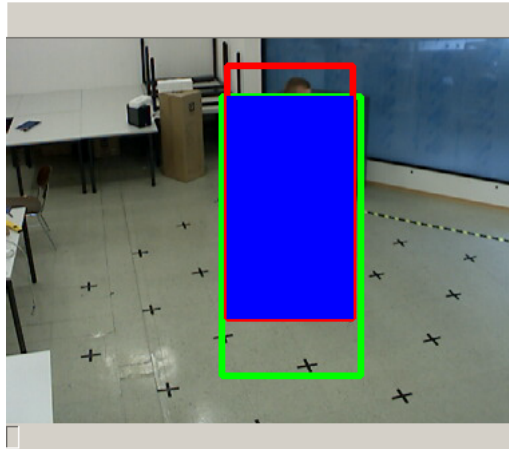


Figura 2: Cálculo del área de solapamiento.

A continuación se puede ver el código para el cálculo del área de solapamiento:

```

if detected_x >= tracker_x:
    inter_x1 = detected_x
else:
    inter_x1 = tracker_x

if detected_y >= tracker_y:
    inter_y1 = detected_y
else:
    inter_y1 = tracker_y

if detected_x + detected_w <= tracker_x + tracker_w:
    inter_x2 = detected_x + detected_w
else:
    inter_x2 = tracker_x + tracker_w

if detected_y + detected_h <= tracker_y + tracker_h:
    inter_y2 = detected_y + detected_h
else:
    inter_y2 = tracker_y + tracker_h

if (inter_x2 - inter_x1) > 0 and (inter_y2 - inter_y1) > 0:
    area = (inter_x2 - inter_x1) * (inter_y2 - inter_y1)
    ratio = area / (detected_w * detected_h)
else:
    area = 0

```

```
ratio = 0
print("Area: ", area, " - Ratio: ", ratio)
```

### 3. Conclusiones

Para realizar las pruebas del script se usó el dataset de la Universidad Tecnológica de Graz, Austria. Si bien se lograron resultados con una tasa de frames aceptable, el tracker no es capaz de realizar un trackeo perfecto de la persona, haciendo que se creen múltiples trackers para un mismo individuo.

Como mejora propuesta, es necesario dotar de un mecanismo mas inteligente al proceso de gestión de los trackers, borrando posibles trackers que estén fuera del encuadre y utilizando técnicas de filtrado para mejorar el trackeo en si.