



VISIÓN ARTIFICIAL PYTHON + OPENCV

SEGMENTACIÓN BASADA EN CAMBIO

PONENTE DIEGO ZAPATA HERNÁNDEZ

GITHUB → [DIEGOZAHE](#)

ÍNDICE

- **Introducción a Python y OpenCV**
- ¿Qué hace nuestro algoritmo?
- Métodos básicos de OpenCV
- Nuestro programa

INTRODUCCIÓN A PYTHON Y OPENCV

PYTHON

Python es:

- Un lenguaje de programación **multiparadigma** (POO, P. imperativa, funcional)
- Un lenguaje **interpretado**.
- De **tipado dinámico**.
- **Multiplataforma**.
- Y lo que mas confunde al principio, es **indentado**

```
# Es un lenguaje interpretado el interprete lee instruccion a instruccion
# Cuando se necesita
def metodo(parametro1, parametro2):
    # Las variables toman un valor
    hola = "hola"
    # Luego podemos cambiar el valor... (tipado dinamico)
    hola = "adios"
```

INTRODUCCIÓN A PYTHON Y OPENCV

OPENCV

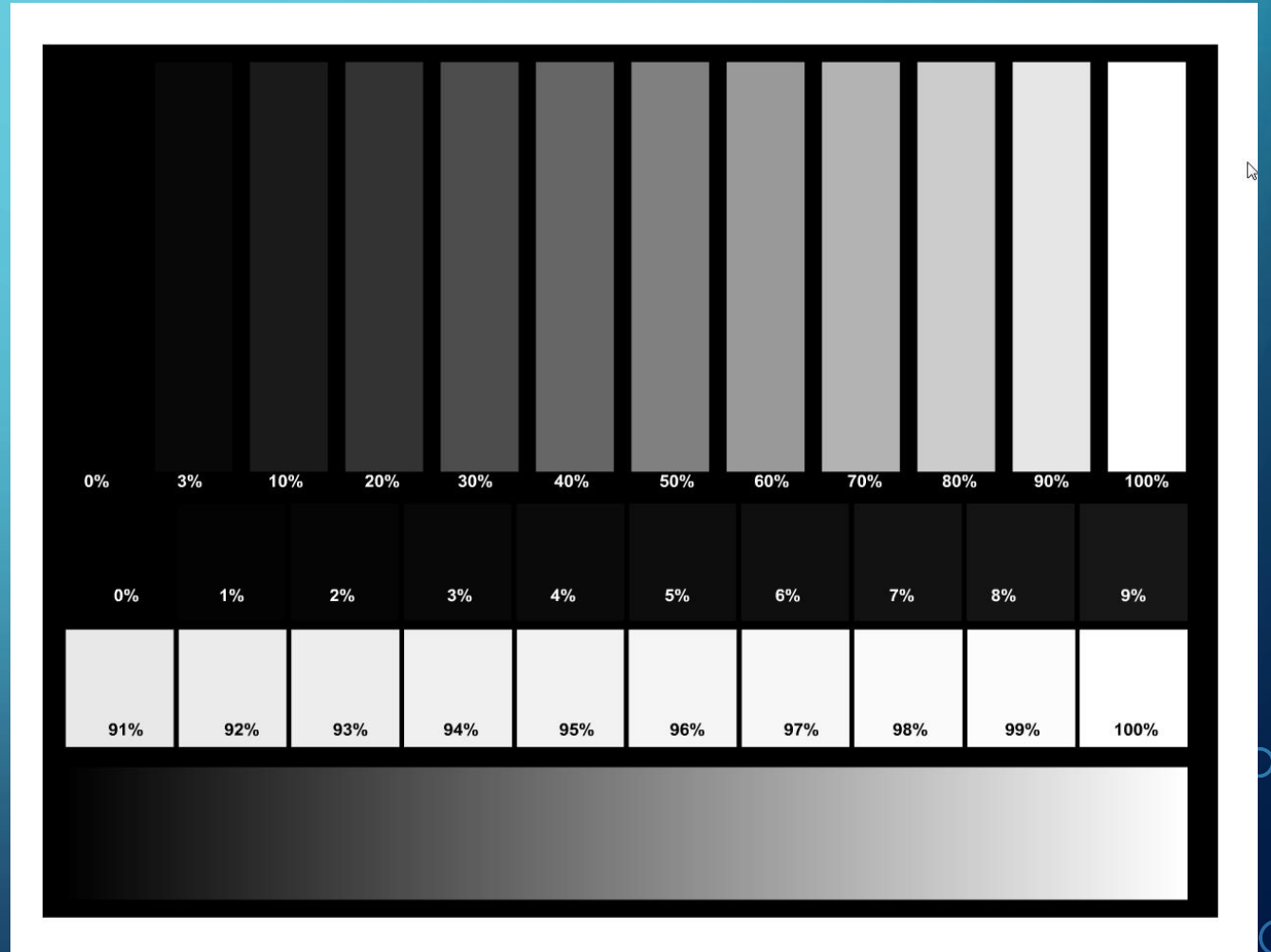
- **OpenCV** es una biblioteca open source para procesamiento de imágenes y visión computarizada.
- Disponible en Linux, Mac, y Windows.
- Tiene estructuras básicas de datos para operaciones con matrices y procesamiento de imágenes.
- En python usamos “NumPy” una biblioteca de funciones matemáticas de alto nivel para las operaciones con vectores o matrices.

ÍNDICE

- Introducción a Python y OpenCV
- **¿Qué hace nuestro algoritmo?**
- Métodos básicos de OpenCV
- Nuestro programa

¿QUÉ HACE NUESTRO ALGORITMO?

- Hace un calculo:
- $\text{ima}(x1,y1)-\text{ima2}(x1,y1)=|r|$



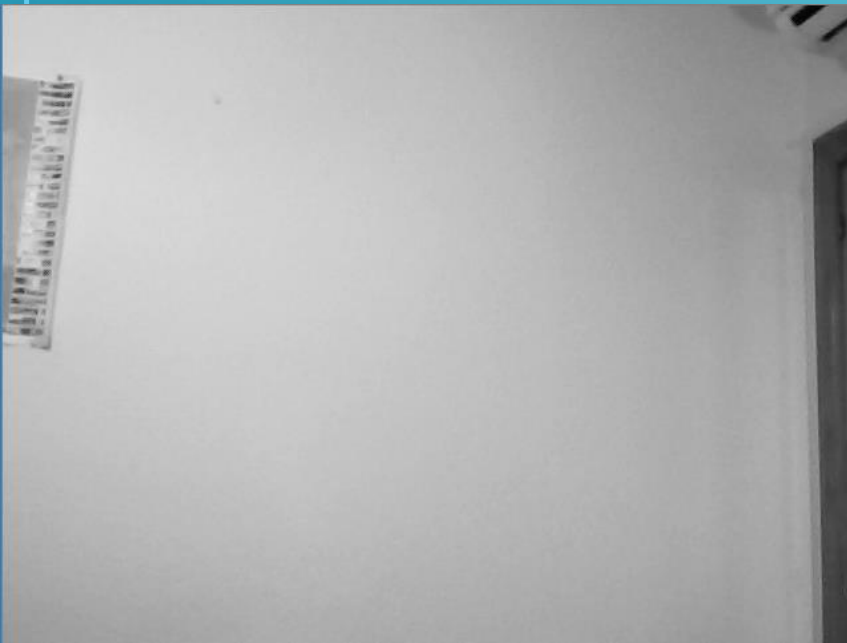
¿QUÉ HACE NUESTRO ALGORITMO?

Comparamos el frame actual con la imagen de referencia para obtener el cambio.

FONDO

Frame con el que comparamos

Resultado de comparar



¿QUÉ HACE NUESTRO ALGORITMO?



ÍNDICE

- Introducción a Python y OpenCV
- ¿Qué hace nuestro algoritmo?
- **Métodos básicos de OpenCV**
- Nuestro programa

MÉTODOS BÁSICOS DE OPENCV

```
import cv2
import numpy as np

# Cargamos el vídeo
camara = cv2.VideoCapture(0)

# Leemos el siguiente frame
(grabbed, frame) = camara.read()

# Mostramos el frame
cv2.imshow("frame", frame)

# Pasar una imagen a color a escala de gris
cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

# Leer una imagen
cv2.imread("ruta_imagen")

# Guardar un frame como imagen.
cv2.imwrite("nombre_imagen.jpg", frame)
```

ÍNDICE

- Introducción a Python y OpenCV
- ¿Qué hace nuestro algoritmo?
- Métodos básicos de OpenCV
- **Nuestro programa**

MÉTODOS IMPORTANTES

```
# Cargamos el vídeo
camara = cv2.VideoCapture(0)
fondo = None
archivoLeido = False

while True:
    #leemos el siguiente frame
    (grabbed, frame) = camara.read()

    #miramos si tenemos el fondo establecido, si no es así lo establecemos
    if archivoLeido == False:
        #creamos la imagen fondo
        archivoLeido = checkfile("fondo.jpg",frame)
        fondo = cv2.cvtColor(cv2.imread("fondo.jpg"), cv2.COLOR_BGR2GRAY)

    # Convertimos a escala de grises
    gris = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Calculo de la diferencia entre el fondo y el frame actual
    resta = cv2.absdiff(fondo, gris)

    # Aplicamos un umbral
    umbral = cv2.threshold(resta, 50, 255, cv2.THRESH_BINARY)[1]

    #hacemos una copia de umbral
    umbralCopy = umbral.copy()

    # Buscamos contorno en la imagen
    im, contornos, hierarchy = cv2.findContours(umbralCopy,cv2.RETR_TREE,cv2.CHAIN_APPROX_NONE)
```

```
def checkfile(archivo,frame):
    try:
        fichero = open(archivo)
        fichero.close()
        print("La imagen existe")
        return True
    except:
        cv2.imwrite("fondo.jpg", frame)
        print ("La imagen se ha creado")
        return True
```

MÉTODOS IMPORTANTES

```
contPersona = 0
# Analizamos cada elemento en la lista
for c in contornos:
    # Eliminamos los contornos más pequeño
    contorno = cv2.contourArea(c)
    # Obtenemos el bounds del contorno, el rectángulo mayor que engloba al contorno
    (x1, y1, w, h) = cv2.boundingRect(c)
    if contorno > 5000:
        # Dibujamos el rectángulo del bounds
        cv2.rectangle(frame, (x1, y1), (x1 + w, y1 + h), (0, 255, 0), 2)
        contPersona += 1
print("Hay", contPersona, "contornos")

# Mostramos el video
cv2.imshow("Camara", frame)
cv2.imshow("Gris", gris)
cv2.imshow("Umbral", cv2.pyrDown(umbral))
cv2.imshow("Resta", cv2.pyrDown(resta))
cv2.imshow("fondo", fondo)
#cv2.imshow("Contorno", cv2.pyrDown(umbralCopy))

# Capturamos una tecla para salir
key = cv2.waitKey(1) & 0xFF

# Si ha pulsado la letra s, salimos
if key == ord("s"):
    break

# When everything done, release the capture
camara.release()
cv2.destroyAllWindows()
```



¿PREGUNTAS?

**VISIÓN ARTIFICIAL
PYTHON + OPENCV**

PONENTE DIEGO ZAPATA HERNÁNDEZ