Universidad Autónoma de San Luis Potosí
Facultad de ingeniería
Tratamiento de Imágenes
Practica 6
Nombre Práctica: Umbralizado y Operaciones Morfológicas
Nombre del Alumno: Manuel Ramírez Galván

Fecha: 11/03/2025

Procedimiento

3.1. Coloque una cámara web en una posición estática. Diseñar un programa en Python en el cual se cumplan los siguientes requisitos:

- Tomar una foto del fondo desde la cámara.
- Sin mover la cámara, poner un objeto en la escena y tomar una segunda foto.
- Convertir estas imágenes a escala de grises.
- Restar las dos imágenes y mostrar el resultado en una ventana.
- Realizar el umbralizado de la imagen restada y mostrar el resultado en otra ventana.
- A la imagen umbralizada realizar las operaciones de erosión, dilatación, apertura y cierre por separado y mostrar los resultados en diferentes ventanas.
- Aplicar la máscara resultante a la segunda imagen capturada.

Resultados

```python
import cv2

cam = cv2.VideoCapture(0)

while (cam.isOpened()):
    ret, frame = cam.read()
    if ret:
        cv2.imshow("Ejercicio 6.1", frame)
        key = cv2.waitKey(1)
        if key == ord('f'):
            cv2.imwrite("Foto1.jpg", frame)
        if key == ord('g'):
            cv2.imwrite("Foto2.jpg", frame)
        if key == ord('q'):
            break
    else:
        break

cam.release()
cv2.destroyAllWindows()
```

Imagen 1.- Código 1 Ejercicio 1

```python
import cv2

img1 = "Foto1.jpg"
img2 = "Foto2.jpg"

img1 = cv2.imread(img1, cv2.IMREAD_GRAYSCALE)
img2 = cv2.imread(img2, cv2.IMREAD_GRAYSCALE)

img1_b = cv2.GaussianBlur(img1, (5, 5), 0)
img2_b = cv2.GaussianBlur(img2, (5, 5), 0)

img_res = cv2.subtract(img1_b, img2_b)

cv2.imshow("Resta", img_res)
cv2.waitKey(0)
cv2.destroyAllWindows()

img_thresh = cv2.threshold(img_res, 95, 255, cv2.THRESH_BINARY)[1]
cv2.imshow("Umbral", img_thresh)
cv2.waitKey(0)
cv2.destroyAllWindows()

k3 = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))
mask1 = cv2.dilate(img_thresh, k3, iterations=1)
cv2.imshow("Dilatacion", mask1)
mask2 = cv2.erode(img_thresh, k3, iterations=1)
cv2.imshow("Erosion", mask2)
mask3 = cv2.morphologyEx(img_thresh, cv2.MORPH_OPEN, k3, iterations=1)
cv2.imshow("Apertura", mask3)
mask4 = cv2.morphologyEx(img_thresh, cv2.MORPH_CLOSE, k3,
cv2.imshow("Cierre", mask4)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Imagen 2.- Código 2 Ejercicio 1

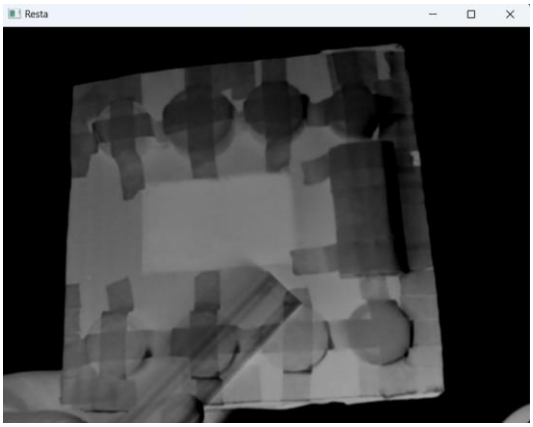

Imagen 3.- Resta de Dos Fotos

Imagen 4.- Máscara

Imagen 5.- Máscara con Erosión

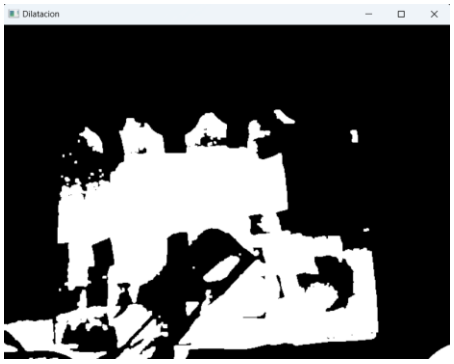

Imagen 6.- Máscara con Dilatación

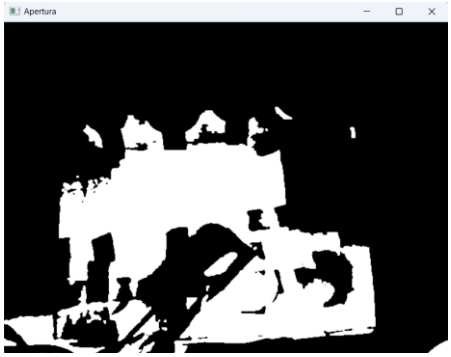

Imagen 7.- Máscara con Apertura

Imagen 8.- Máscara con Cierre

Procedimiento

3.2. Diseñe un programa en Python en el cual se cumplan los siguientes requisitos:

- Cargar un video del disco duro.
- Convertir los fotogramas a escala de grises.
- Restar fotograma actual con fotograma anterior tomando su resta de valor absoluto.
- Realizar el umbralizado de la imagen restada.
- Aplicar un dilatado a la imagen umbralizada.
- Encuentra las áreas que ha cambiado de fotograma en fotograma (Usar cv2.findContours).
- Encapsular en un cuadrado las áreas que han cambiado de fotograma en fotograma.

Resultados

```python
import cv2

def deteccion_movimiento(curr, prev):
    diff_frame = cv2.absdiff(curr, prev)
    cv2.imshow("Video", diff_frame)

    thresh_frame = cv2.threshold(diff_frame, 30, 255, cv2.THRESH_BINARY)[1]

    kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))
    diff_frame = cv2.morphologyEx(diff_frame, cv2.MORPH_OPEN, kernel, iterations=1)

    cv2.imshow("Video", thresh_frame)
    return diff_frame

vid_capture = cv2.VideoCapture("C:/Users/HUAWEI/Desktop/L TRATAMIENTO/Manual/Practica 6 - Umbralizacion y
operaciones morfologicas/Recursos/Calle.mp4")

prev_frame = None

while(vid_capture.isOpened()):
    ret, frame = vid_capture.read()
    if ret:
        key = cv2.waitKey(10)
        if key == ord('q'):
            break
        prep_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        prep_frame = cv2.GaussianBlur(prep_frame, (3, 3), 0)

        if prev_frame is None:
            prev_frame = prep_frame
            continue

        mov_frame = deteccion_movimiento(prep_frame, prev_frame)
        prev_frame = prep_frame

        contours, _ = cv2.findContours(mov_frame, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        cv2.drawContours(frame, contours, -1, (0, 255, 0), 2, cv2.LINE_AA)

        cv2.imshow("Video", frame)

    else:
        break

vid_capture.release()
cv2.destroyAllWindows()
```

Imagen 9.- Código Ejercicio 2

Imagen 10.- Video con Detección de Movimiento

Comprensión

1. **¿Cuáles son los tipos de umbralizado y define cada uno?**

Umbralizado Simple o Binario: Convierte los pixeles en blanco (255) o negro (0) según un umbral fijo. Si el valor del píxel es mayor que el umbral, se convierte en blanco y si el valor del píxel es menor o igual al umbral, se convierte en negro.

Umbralizado Simple Invertido: Similar al umbralizado simple, pero invierte los colores. Si el valor del píxel es mayor que el umbral, se convierte en negro y si el valor del píxel es menor o igual al umbral, se convierte en blanco.

Umbralizado de Truncamiento: Si el valor del píxel supera el umbral, se reemplaza por el valor del umbral; de lo contrario, se mantiene igual.

Umbralizado Adaptativo: Ajusta el umbral localmente en diferentes regiones de la imagen.

Umbralizado de Otsu: Calcula automáticamente el mejor umbral basado en la distribución de los píxeles.

Umbralizado a Cero: Solo conserva los píxeles mayores que el umbral, convirtiendo el resto en negro.

Umbralizado a Cero Invertido: Funciona como el umbralizado a cero, pero los valores por encima del umbral se vuelven 0 y los por debajo se conservan.

2. **¿Para qué se utilizan las operaciones morfológicas?**

Las operaciones morfológicas son técnicas de procesamiento de imágenes aplicadas en imágenes binarias o en escala de grises que usan una estructura para modificar la forma de los objetos en la imagen.

Se utilizan principalmente para:

- Eliminación de ruido en imágenes segmentadas.
- Mejorar la detección de contornos en imágenes de visión artificial.
- Refinamiento de regiones en imágenes procesadas.
- Rellenar huecos o separar objetos conectados en segmentación de imágenes.
- Preprocesamiento en reconocimiento de patrones, como en detección de caracteres o en visión computacional.

3. **¿Cuáles son las operaciones morfológicas?**

Erosión: Elimina pixeles en los bordes de los objetos, hace que los objetos se hagan mas pequeños y puede separar objetos conectados.

Dilatación: Expande los bordes de los objetos, aumentando su tamaño, puede rellenar huecos y reforzar estructuras en imágenes segmentadas.

Apertura (Erosión + Dilatación): Elimina ruido sin afectar la forma de los objetos.

Cierre (Dilatación + Erosión): Rellena huecos pequeños dentro de objetos de los objetos sin afectar el tamaño general.

Conclusiones

El umbralizado permite segmentar objetos al convertir imágenes a binario mediante un umbral, existen diferentes tipos como el simple, adaptativo, truncado, los invertidos y el Otsu, cada uno tiene su diferente función, lo que ayuda a hacer un buen umbralizado en esta practica es tener un fondo uniforme y una buena iluminación para evitar ruido y objetos no deseados.

Las operaciones morfológicas ayudan a mejorar las imágenes segmentadas, eliminado ruido, resaltando contornos y mejorando la estructura de los objetos.