Universidad Autónoma de San Luis Potosí
Facultad de ingeniería
Tratamiento de Imágenes
Practica 7
Nombre Práctica: Detección de Contornos
Nombre del Alumno: Manuel Ramírez Galván

Fecha: 18/03/2025

Procedimiento

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

- Convertir estas imágenes a escala de grises y umbralizarla para separarla del fondo.
- Encontrar los contornos de las monedas.
- En la imagen original, dibujar los contornos encontrados y el centroide para cada uno.
- En la imagen umbralizada, el área para cada moneda.
- En la consola, se debe imprimir el número de monedas que se encuentran en la imagen.

Resultados

```python
import cv2
import numpy as np
import matplotlib.pyplot as plt

img_path = "C:/Users/HUAWEI/Desktop/L TRATAMIENTO/Manual/Practica 7 - Deteccion de
contornos/Imagenes/Monedas.png"
img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
img_color = cv2.imread(img_path, cv2.IMREAD_COLOR)

img_blurr = cv2.GaussianBlur(img, (5, 5), 0)
_, thersh = cv2.threshold(img_blurr, 10, 255, cv2.THRESH_BINARY)

kernel = np.ones((5, 5), np.uint8)
morph = cv2.morphologyEx(thersh, cv2.MORPH_CLOSE, kernel, iterations=1)

contours, _ = cv2.findContours(morph, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

num_monedas = len(contours)
print(f"Numero de monedas: {num_monedas}")

img_countours = img_color.copy()
cv2.drawContours(img_countours, contours, -1, (0, 255, 0), 2)

for contour in contours:
    M = cv2.moments(contour)
    if M["m00"] != 0:
        cx = int(M["m10"] / M["m00"])
        cy = int(M["m01"] / M["m00"])
        cv2.circle(img_countours, (cx, cy), 5, (0, 0, 255), -1)

plt.figure(figsize=(15, 15))
plt.subplot(121)
plt.imshow(img, cmap="gray")
plt.title("Imagen Original")
plt.axis("off")

plt.subplot(122)
plt.imshow(cv2.cvtColor(img_countours, cv2.COLOR_BGR2RGB))
plt.title("Contornos y Centroides")
plt.axis("off")

plt.show()
```

Imagen 1.- Código 1 Ejercicio 1

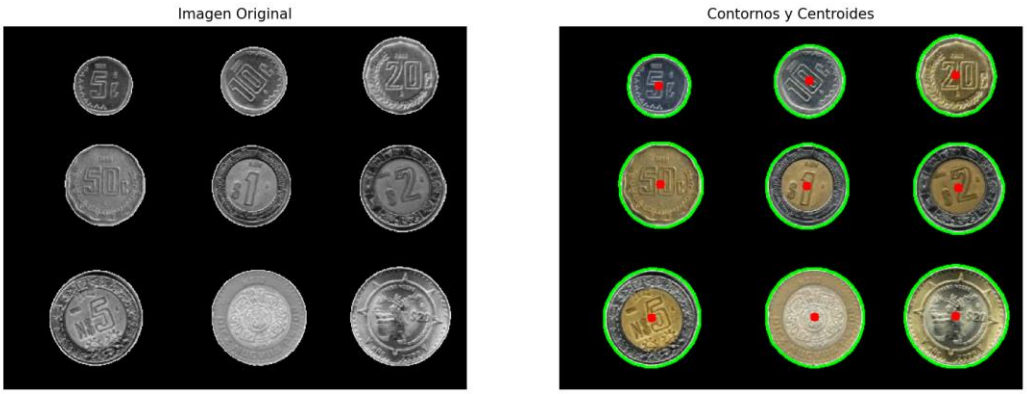

Imagen 2.- Detección de Contornos y Centroides

Procedimiento

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

- Cargar una imagen de distintas monedas desde el disco duro.
- Convertir estas imágenes a escala de grises y umbralizarla para separarla del fondo.
- Encontrar los contornos de las monedas.
- En la imagen original, dibujar los contornos encontrados.
- En base a las áreas encontradas en el ejercicio anterior, crear un diccionario para el área de cada moneda en relación con su valor monetario.
- Denominar cada moneda con su valor monetario en base a su área.
- Mostrar una suma del valor monetario total que hay en la imagen.

Resultados

```python
import cv2
import numpy as np
import matplotlib.pyplot as plt

img_path = "C:/Users/HUAWEI/Desktop/L TRATAMIENTO/Manual/Practica 7 - Deteccion de
contornos/Imagenes/Monedas.png"
img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
img_color = cv2.imread(img_path, cv2.IMREAD_COLOR)

img_blurr = cv2.GaussianBlur(img, (5, 5), 0)

_, thersh = cv2.threshold(img_blurr, 10, 255, cv2.THRESH_BINARY)

kernel = np.ones((5, 5), np.uint8)
morph = cv2.morphologyEx(thersh, cv2.MORPH_CLOSE, kernel, iterations=1)

contours, _ = cv2.findContours(morph, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

valores_monedas = {
    (3500, 4500): 0.05,   # Moneda de 5 centavos
    (4500, 6000): 0.10,   # Moneda de 10 centavos
    (6000, 6500): 0.20,   # Moneda de 20 centavos
    (7000, 8000): 0.50,   # Moneda de 50 centavos
    (6500, 7000): 1.0,    # Moneda de 1 peso
    (8000, 9000): 2.0,    # Moneda de 2 pesos
    (9000, 9600): 5.0,    # Moneda de 5 pesos
    (9600, 10000): 10.0,  # Moneda de 10 pesos
    (10000, 11000): 20.0  # Moneda de 20 pesos
}

monedas_detectadas = []
valor_total = 0

img_countours = img_color.copy()
cv2.drawContours(img_countours, contours, -1, (0, 255, 0), 2)

for contour in contours:
    M = cv2.moments(contour)
    if M["m00"] != 0:
        cx = int(M["m10"] / M["m00"])
        cy = int(M["m01"] / M["m00"])

        area = cv2.contourArea(contour)

        valor_moneda = 0
        for (min_area, max_area), valor in valores_monedas.items():
            if min_area <= area < max_area:
                valor_moneda = valor
                break

        monedas_detectadas.append({"centroide": (cx, cy), "area": area, "valor": valor_moneda})
        valor_total += valor_moneda

        cv2.circle(img_countours, (cx, cy), 5, (0, 0, 255), -1)

        cv2.putText(img_countours, f"${valor_moneda}", (cx - 20, cy - 10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 2)

print(f"Cantidad de monedas detectadas: {len(monedas_detectadas)}")
print(f"Valor total en la imagen: ${valor_total}")

plt.figure(figsize=(15, 15))
plt.subplot(121)
plt.imshow(img, cmap="gray")
plt.title("Imagen Original")
plt.axis("off")

plt.subplot(122)
plt.imshow(cv2.cvtColor(img_countours, cv2.COLOR_BGR2RGB))
plt.title(f"Contornos y Valores - Total: ${valor_total}")
plt.axis("off")

plt.show()
```

Imagen 3.- Código Ejercicio 2

Imagen 4.- Reconocimiento de Valor de Monedas y Suma Total del Valor

Comprensión

1. **¿Para qué se utiliza la detección de contornos?**

Para identificar los bordes de una imagen, por ejemplo:

- Segmentación de objetos del fondo para su análisis individual.
- Reconocimiento de formas y patrones, como figuras geométricas o símbolos.
- Medición de características como área, perímetro, centroide.
- Seguimiento de objetos.

2. **¿Para qué se utilizan los momentos en la detección de contornos?**

Los momentos son medidas matemáticas que describen la forma, tamaño y orientación de un objeto en una imagen a partir de sus contornos, pueden cuantificar propiedades geométricas de un contorno. Se usan para encontrar el centro, calcular el área, orientación y comparar formas, son útiles para la visión artificial y reconocimiento de objetos.

3. **¿Para qué se utilizan el área en la detección de contornos?**

- Filtrar objetos por tamaño al destacar contornos muy pequeños, el cual son el ruido, o muy grandes, que son el fondo.
- Clasificar formas según su tamaño para agrupar objetos y distinguir diferentes tipos de objetos.
- Medir características físicas o proporciones, esto sirve si se conoce el tamaño real de los pixeles y estimar el área real.
- Detectar y seguir objetos en movimiento ya que puede cambiar el área si el objeto se mueve o cambia de forma.

Conclusiones

La identificación de contornos nos sirve para identificar los bordes de un objeto de una imagen el cual podemos segmentar y analizar formas, medir áreas, perímetros, detección de bordes, etc.

En este caso se utilizaron las áreas para asignarle un rango a cada una y poder darle su valor monetario a cada moneda y poder contar que monedas hay, de que tipo y el total monetario esto con una sola fotografía, esto puede servir para una mayor cantidad de monedas en donde la fotografía se tome en la misma posición y distancia de las monedas.