Reporte de Práctica: Eliminación de Ruido con Operaciones Morfológicas TopHat y BlackHat

1. Objetivo

Aplicar técnicas morfológicas de procesamiento de imágenes para eliminar ruido de una máscara de detección de colores en imágenes capturadas por cámara. Se utiliza la combinación de las operaciones morfológicas TopHat y BlackHat para mejorar la calidad de la máscara y así obtener una detección positiva (F+) y negativa (F-) más limpia y precisa.

---

2. Fundamento Teórico

En visión artificial, las imágenes capturadas suelen contener ruido debido a iluminación irregular, reflejos, sombras o interferencias en la adquisición. El ruido dificulta la correcta detección y segmentación de objetos.

Las operaciones morfológicas son herramientas fundamentales para procesar imágenes binarias (máscaras), mejorando la detección:

- **TopHat**: destaca pequeñas regiones claras que son más brillantes que su entorno inmediato. Ayuda a resaltar detalles que podrían perderse o a corregir imperfecciones en la máscara.
- **BlackHat**: destaca pequeñas regiones oscuras que son más oscuras que su entorno. Es útil para identificar y eliminar huecos o puntos negros causados por ruido.

Aplicando estas operaciones y combinándolas adecuadamente, se puede limpiar la máscara eliminando el ruido y resaltando los detalles importantes para una detección más fiable.

---

3. Descripción del Código

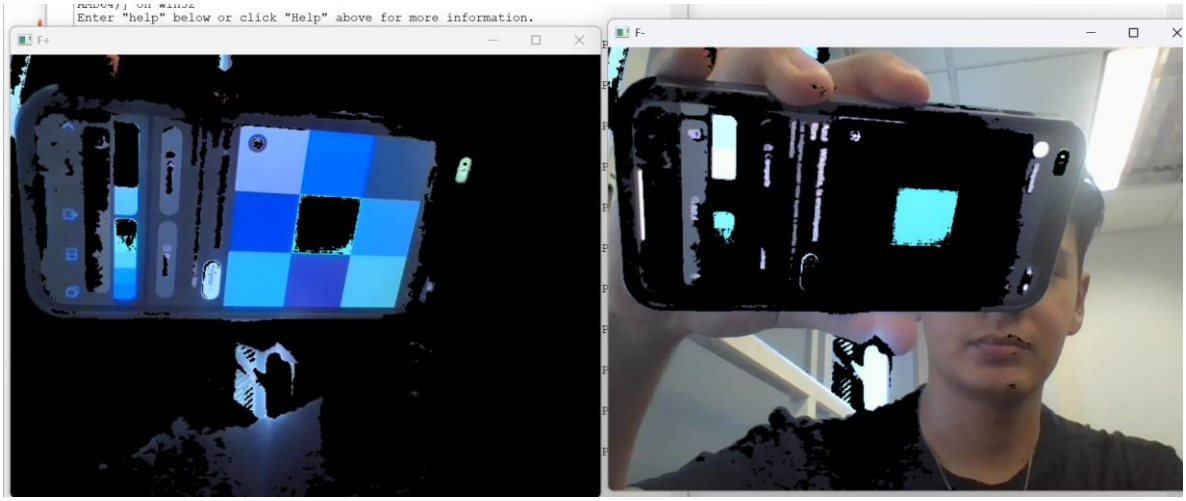
1. **Captura de imagen:** Se captura un video en tiempo real usando la cámara del equipo.
2. **Conversión a HSV:** Se convierte cada frame de BGR a HSV para facilitar la detección de colores específicos.

3. **Definición de máscaras de color:** Se definen rangos HSV para detectar rojo, verde y azul, creando máscaras binarias para cada color.
4. **Combinación de máscaras:** Se unen las máscaras de los tres colores para trabajar con una máscara compuesta.
5. **Aplicación de TopHat y BlackHat:**
    - Se aplica TopHat para resaltar regiones claras pequeñas.
    - Se aplica BlackHat para resaltar regiones oscuras pequeñas.
6. **Limpieza de la máscara:**
    - Se resta BlackHat para eliminar puntos oscuros de ruido.
    - Se suma TopHat para recuperar detalles importantes claros.
7. **Obtención de F+ y F-:**
    - F+ es la imagen original con la máscara limpia aplicada, mostrando solo los colores detectados sin ruido.
    - F- es el complemento, mostrando todo lo demás.
8. **Visualización:** Se muestran en ventanas separadas las imágenes F+ y F-.

---

4. Conclusiones

- La combinación de operaciones morfológicas TopHat y BlackHat es efectiva para mejorar la calidad de las máscaras binarias, ayudando a eliminar ruido y preservar detalles relevantes.
- Este método permite obtener una segmentación más precisa en aplicaciones de detección de color en tiempo real.
- Se recomienda ajustar el tamaño del kernel morfológico según la escala del ruido y las características de la imagen para mejores resultados.



Código:

```
import cv2

import numpy as np


cap = cv2.VideoCapture(0)


# Kernel morfológico

kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (5, 5))


while True:

  ret, frame = cap.read()

  if not ret:

    break


  hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)


  # Rango para rojo

  lower_red1 = np.array([0, 120, 70])
```

```
    upper_red1 = np.array([10, 255, 255])
    lower_red2 = np.array([170, 120, 70])
    upper_red2 = np.array([180, 255, 255])
    mask_red = cv2.inRange(hsv, lower_red1, upper_red1) | cv2.inRange(hsv,
lower_red2, upper_red2)

    # Rango para verde
    lower_green = np.array([36, 50, 70])
    upper_green = np.array([89, 255, 255])
    mask_green = cv2.inRange(hsv, lower_green, upper_green)

    # Rango para azul
    lower_blue = np.array([94, 80, 2])
    upper_blue = np.array([126, 255, 255])
    mask_blue = cv2.inRange(hsv, lower_blue, upper_blue)

    # Máscara combinada
    combined_mask = cv2.bitwise_or(mask_red, mask_green)
    combined_mask = cv2.bitwise_or(combined_mask, mask_blue)

    # 🔍 Aplicamos TopHat y BlackHat a la máscara para limpiar detalles
    tophat = cv2.morphologyEx(combined_mask, cv2.MORPH_TOPHAT, kernel)
    blackhat = cv2.morphologyEx(combined_mask, cv2.MORPH_BLACKHAT, kernel)

    # 🧼 Limpiamos la máscara combinando resultados
    cleaned_mask = cv2.subtract(combined_mask, blackhat)
```

```
    cleaned_mask = cv2.add(cleaned_mask, tophat)

    # F+ (detección positiva)
    F_pos = cv2.bitwise_and(frame, frame, mask=cleaned_mask)

    # F- (resto de la imagen)
    mask_inv = cv2.bitwise_not(cleaned_mask)
    F_neg = cv2.bitwise_and(frame, frame, mask=mask_inv)

    # Mostrar solo F+ y F-
    cv2.imshow('F+', F_pos)
    cv2.imshow('F-', F_neg)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```