VISIÓN ARTIFICIAL

Práctica 5. Umbralización

Ingeniería en Mecatrónica
6to semestre

Mtro. Mauricio Alejandro Cabrera Arellano

Arturo Alejandro Guzman Perez 22110356 6G

Práctica 5. Umbralización

Objetivo

Aplicar técnicas de umbralización en imágenes utilizando la biblioteca OpenCV en Python, con el fin de simplificar datos visuales para su análisis, mediante la conversión de imágenes a formatos binarios que faciliten la detección y procesamiento de elementos clave.

La umbralización es una técnica de procesamiento de imágenes que convierte una imagen en escala de grises en una imagen binaria. Esto se logra asignando valores de blanco o negro a los píxeles, dependiendo de si su intensidad supera o no un valor umbral determinado. Esta simplificación es útil para resaltar características específicas de una imagen y facilitar su análisis.

Existen diferentes métodos de umbralización:

- **Umbralización binaria simple:** Asigna un valor máximo (por ejemplo, 255) a los píxeles cuya intensidad supera el umbral y cero a los demás.
- **Umbralización adaptativa:** Calcula diferentes umbrales para distintas regiones de la imagen, lo que es útil en condiciones de iluminación no uniformes.
- **Umbralización de Otsu:** Determina automáticamente el umbral óptimo minimizando la varianza intra-clase, ideal para imágenes con histogramas bimodales.

Desarrollo de la Práctica

- *Lectura de la Imagen*

Se carga la imagen a procesar:

```
import cv2
import numpy as np

img = cv2.imread('bookpage.jpg')
```

- *Umbralización Binaria Simple*

Se aplica un umbral fijo a la imagen en color:

```
retval, threshold = cv2.threshold(img, 12, 255, cv2.THRESH_BINARY)
cv2.imshow('Threshold', threshold)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Este método convierte los píxeles con valores superiores a 12 en 255 (blanco) y los demás en 0 (negro).

- *Conversión a Escala de Grises y Umbralización*

Se convierte la imagen a escala de grises antes de aplicar el umbral:

```
grayscaled = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
retval, threshold = cv2.threshold(grayscaled, 10, 255, cv2.THRESH_BINARY)
cv2.imshow('Threshold Grayscale', threshold)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

La conversión a escala de grises simplifica la imagen, facilitando la aplicación de umbrales.

- *Umbralización Adaptativa*

Se aplica un umbral que varía según la región de la imagen:

```
th = cv2.adaptiveThreshold(grayscaled, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                           cv2.THRESH_BINARY, 115, 1)
cv2.imshow('Adaptive Threshold', th)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Este método es útil para imágenes con iluminación desigual, ya que calcula el umbral para pequeñas regiones de la imagen.

- *Umbralización de Otsu*

Se utiliza el método de Otsu para determinar automáticamente el umbral óptimo:

python

```
retval2, threshold2 = cv2.threshold(grayscaled, 125, 255,
                                    cv2.THRESH_BINARY + cv2.THRESH_OTSU)
cv2.imshow('Otsu Threshold', threshold2)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Este método es eficaz cuando el histograma de la imagen presenta dos picos distintos.

4. Resultados

- **Umbralización binaria simple:** La imagen resultante muestra una separación básica entre áreas claras y oscuras, pero puede no ser suficiente en condiciones de iluminación variables.
- **Conversión a escala de grises y umbralización:** Mejora la claridad de la imagen, pero aún puede presentar problemas en áreas con sombras o iluminación desigual.
- **Umbralización adaptativa:** Proporciona una mejor separación en imágenes con iluminación no uniforme, resaltando detalles que podrían perderse con un umbral fijo.
- **Umbralización de Otsu:** Determina automáticamente el umbral óptimo, siendo eficaz en imágenes con histogramas bimodales, pero menos efectiva en imágenes con iluminación desigual.

Conclusión

La umbralización es una técnica fundamental en el procesamiento de imágenes que permite simplificar y resaltar características importantes para su análisis. La elección del método de umbralización adecuado depende de las características de la imagen:

- **Umbralización binaria simple:** Adecuada para imágenes con iluminación uniforme y contraste claro entre objetos y fondo.
- **Umbralización adaptativa:** Recomendable para imágenes con iluminación no uniforme, ya que ajusta el umbral localmente.
- **Umbralización de Otsu:** Útil cuando se desconoce el umbral óptimo y la imagen presenta un histograma bimodal.

El uso de OpenCV en Python facilita la implementación de estos métodos, proporcionando herramientas eficientes para el procesamiento y análisis de imágenes.