# PRACTICA 3

*Sara De Los Ríos Colorado Juan Esteban Zapata Espinal*

Ingeniería Mecatrónica Facultad de ingeniería

Instituto Tecnológico Metropolitano [Sarade259367@correo.itm.edu.co](mailto:Sarade259367@correo.itm.edu.co) [Juanzapata249444@correo.itm.edu.co](mailto:Juanzapata249444@correo.itm.edu.co)

31 de Marzo de 2022 Medellín

# RESUMEN

# Las operaciones morfológicas son una herramienta importante al momento de hablar de visión artificial, ya que brindan soluciones para eliminar ruido en imágenes binarizadas, estas se refieren a operaciones simples que se expanden en función de la forma de la imagen y son de gran ayuda para resaltar detalles que a simple vista pueden ser imperceptibles al ojo humano. Por otro lado, gran parte de la responsabilidad del éxito de la aplicación de dichas operaciones, recae sobre el tipo de iluminación que se le aplique a la imagen; la iluminación define que detalles se van a observar de la imagen, una correcta iluminación podría ser la base para resaltar información que inicialmente no se podía observar fácilmente, y por el contrario, una mala iluminación puede provocar grandes pérdidas de información. Por medio de Python, se llevaron a cabo diferentes procesos de operaciones morfológicas, jugando con varios niveles de iluminación y segmentación dependiendo de la aplicación. Adicionalmente, se llevó a cabo por medio de segmentación, varios cambios de componentes de color a diferentes colores.

# PALABRAS CLAVE

OpenCv, iluminación, contornos, máscara, segmentación, operadores morfológicos

# INTRODUCCIÓN

# En algunas ocasiones, al aplicar el procesamiento de imágenes puede pasar que el interés recaiga sobre una zona u objeto en específico. Cuando esto pasa, se puede recurrir a la segmentación y operaciones morfológicas para definir cual es la zona u objeto de interés, y posteriormente realizar el análisis respectivo. Esta práctica tiene como objetivo implementar por medio de funciones de OpenCv y numpy diferentes métodos para segmentar imágenes y detectar bordes. Además, se implementaron operaciones morfológicas para simplificar y corregir las formas de los objetos en las imágenes, dependiendo de la aplicación que se estuviera llevando a cabo.

# PROCEDIMIENTO

**Punto A**

Es fundamental tener en cuenta la iluminación en los sistemas de visión artificial, con una buena iluminación el sistema podrá trabajar con mayor facilidad, por esto existen diferentes técnicas, las más comunes son:

[1] **Luz frontal:** la cámara se ubica mirando al objetivo en la misma dirección de la luz tal y como se observa en la figura 1, reduciendo así las sombras y suavizando texturas. Sus ventajas son la eliminación de sombras y que puede utilizarse a grandes distancias, como desventaja tenemos que produce grandes reflejos en superficies reflectantes.

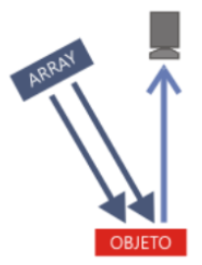
Se aplica en objetos tales como papel, tela y demás superficies mate para detección de diferentes colores y cambios que supongan un cambio de color en la superficie.



*Figura 1. Luz frontal*

**Luz lateral:** la cámara se ubica mirando al objetivo mientras que la luz viene desde un lateral como se aprecia en la figura 2, el grado de inclinación será de acuerdo al grado deseado de resalte de los relieves. Como ventajas tenemos que realza los pequeños relieves de los objetos, pero a su vez si se usa un ángulo muy pequeño con respecto a la horizontal la luz producirá sombras de todos los relieves.

Es ideal para resaltar bordes, rayas y fisuras en una dirección determinada.



*Figura 2. Luz lateral.*

**Iluminación por campo oscuro (Darkfield):** La luz es emitida lateralmente con un ángulo muy pequeño mediante un anillo en todas las direcciones como vemos en la figura 3, rebotando en los defectos del objeto a analizar e incidiendo en la cámara. Sus ventajas son que destaca los detalles en superficies con poco contraste, pero no es recomienda para superficies que absorban luz.

Es perfecta para resaltar incrustaciones y códigos alfanuméricos con poco contraste en metal sobre metal o gris sobre gris. Como grabados laser o troquel.



*Figura 3.* *Darkfield.*

**Iluminación por contraste (Backlight):** la luz se emite des la parte de atrás del objeto tal y como se muestra en la figura 4, dejándolo entre la fuente de iluminación y el cámara, por esta razón debe de ser uniforme y la cámara inspecciona la silueta del objeto pudiendo realizar medidas de alta precisión e impurezas en objetos traslucidos, pero no permite reconocer detalles superficiales.

Se usa para la inspección de la silueta del objeto y detectar manchas rayas o gritas en materiales traslucidos.



*Figura 4.* *Backlight.*

**Iluminación Axial difusa:** La luz es emitida lateralmente siendo reflejada 90º por un espejo semitransparente que desvía los haces de luz en la misma dirección que el eje de la cámara tal y como se muestra en la figura 5, consiguiendo una luz difusa homogénea. Esto nos facilita la inspección de códigos en materiales altamente reflectivos, pero no permite reconocer relieves en el objeto.

es la indicada para la inspección de superficies reflectantes como PCB, etiquetas reflectantes o cavidades profundas.



*Figura 5. Axial difusa.*

**Iluminación difusa Tipo Domo:** La luz es emitida dentro de una cúpula esférica resultando una luz difusa desde todas direcciones como se muestra en la figura 6, eliminando sombras y reflejos, suavizando texturas y minimizando la influencia de rayas, polvo, relieves y curvaturas que pueda tener el objeto inspeccionado. Esto nos da como ventaja la eliminación de sombras y minimizar arrugas y relieves trayendo como única contraparte su elevando costo.

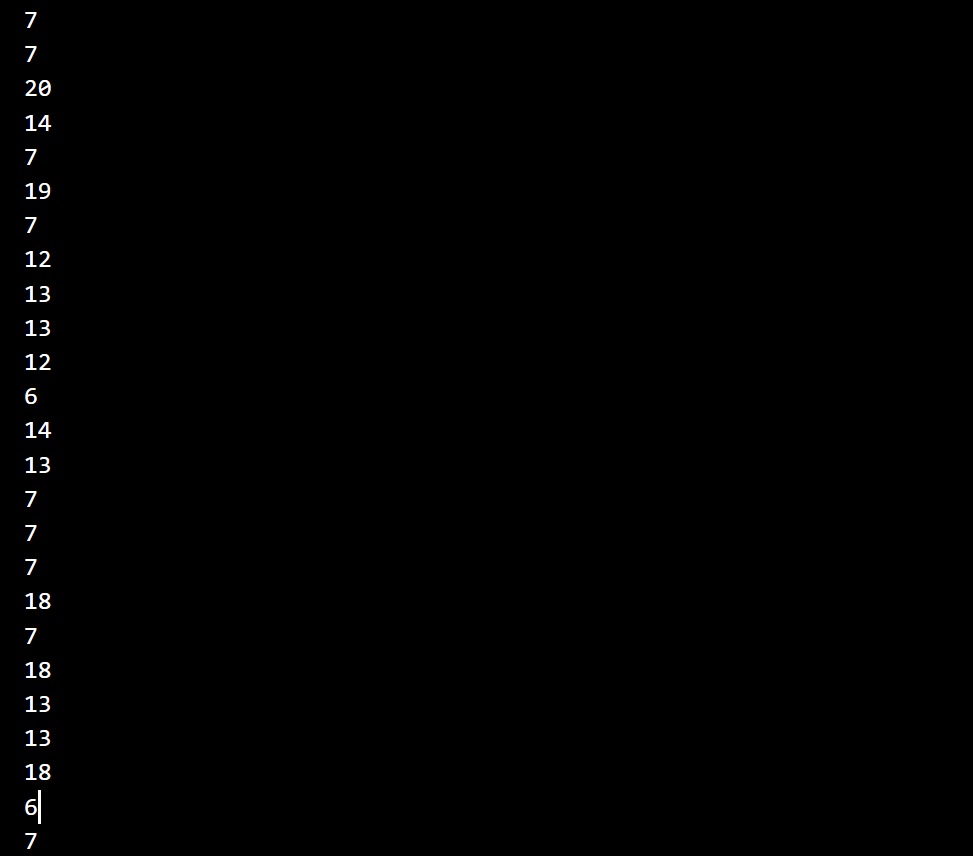
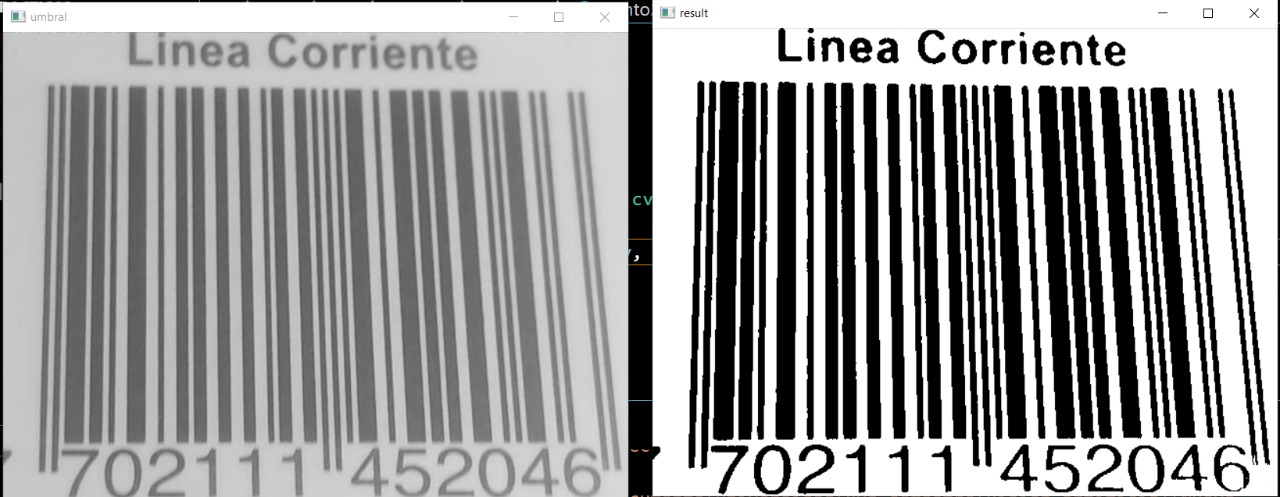
Es especialmente buena para la inspección de instrumental médico, espejos, latas, entre otros



*Figura 6. Difusa tipo domo.*

**Punto B**

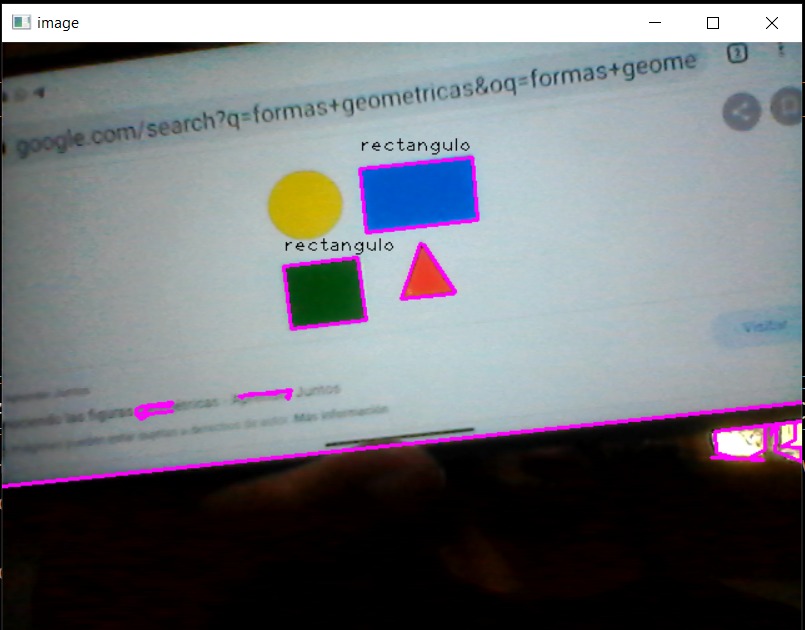
[2] La umbralización es una técnica de segmentación ampliamente utilizada en las aplicaciones industriales. Se emplea cuando hay una clara diferencia entre los objetos a extraer respecto al fondo de la escena. Los principios que rigen son la similitud entre los píxeles pertenecientes a un objeto y sus diferencias respecto al resto. Por lo tanto, la escena debe caracterizarse por un fondo uniforme y por objetos parecidos. [3] Se considera el mejor algoritmo para la selección de umbrales en la segmentación de imágenes, es simple de calcular y no se ve afectado por el brillo y el contraste de la imagen, por lo que se ha utilizado ampliamente en el procesamiento de imágenes digitales. Divide la imagen en dos partes, el fondo y el primer plano, según las características grises de la imagen. Dado que la varianza es una medida de la uniformidad de la distribución de la escala de grises, cuanto mayor sea la varianza entre clases entre el fondo y el primer plano, mayor será la diferencia entre las dos partes de la imagen. Cuando una parte del primer plano se divide incorrectamente en el fondo o una parte del fondo se divide incorrectamente en el primer plano hará que la diferencia entre las dos partes se haga más pequeña. Para una óptima selección del umbral se debe maximizar la varianza entre clases. Su desventaja principal es que es sensible al ruido de la imagen; solo se puede segmentar para un solo objetivo; cuando la relación de tamaño del objetivo y del fondo es grande y la función de variación entre clases puede mostrar picos dobles o picos múltiples, el efecto no es bueno en este momento.

**Para llevar a cabo la lectura de un código de barras seleccionado, se requiere en primer lugar binarizar la imagen, es decir, que esté compuesta únicamente por los colores negro y blanco, a partir de esta se lleva a cabo la lectura de los pixeles de izquierda a derecha manteniendo constante en valor en [i], de este modo, se leerá el color blanco y no se imprimirá ningún valor hasta que este se encuentre con un pixel negro y continue contando los demás pixeles de este mismo color hasta que se encuentre con otro blanco y finalmente imprimir el resultado, ver Figura 7.

*Figura 7. Primeros 20 valores impresos del conteo de pixeles y segmentación con luz frontal.*

# Punto C

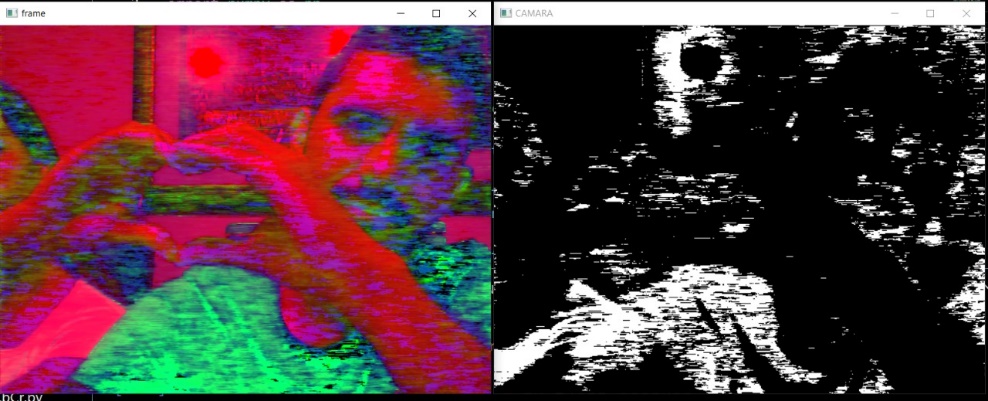
Los contornos se pueden definir como la línea que une todos los puntos a lo largo de un límite de una imagen los cuales manejan la misma intensidad. Son útiles para analizar formas, tamaños y detección de objetos. Para encontrar los contornos de una imagen existe un algoritmo propio de OpenCv, FindContours, el cual solamente acepta los parámetros de imágenes binarias, así que para poder usar esta función se requiere convertir en primer lugar la imagen leída a escala de grises y luego a una binaria. Esta función devuelve dos valores, uno es el contorno mismo y el otro es la propiedad correspondiente a cada contorno, jerarquía. Dicha función está compuesta por los siguientes parámetros cv2.findContours(image, mode, method); el primer parámetro es la imagen que busca el contorno, el segundo representa el modo de recuperación del contorno y el tercero es el método de aproximación del contorno. En el ejemplo que se realizó se detectan los contornos correspondientes a rectángulos, ver Figura 8.



*Figura 8.* *Detección de bordes usando cv2.findContours*

**Punto D**

Para segmentar correctamente los 7 colores, primero debemos transformar la imagen de BGR a HSV para luego determinar los rangos en donde se encuentran los colores a segmentar del cual cabe resaltar que el rango de tono (H) es [0,179], el rango de saturación (S) es [0,255] y el rango de valor (V) es [0,255]. Al tener estas mascaras definidas se procede a aplicarlas a la imagen capturada, ver Figura 10.

****

*Figura 10. Imagen segmentada en HSV*

**Punto E**

**Punto F**

# CONCLUSIONES

* La iluminación toma un factor muy importante a la hora de aplicar la segmentación, puesto que una incorrecta iluminación puede generar dificultades y, por ende, pérdida de información.
* Al implementar el método de segmentación de Otsu, se logra observar que este arroja una mejor calidad de la imagen segmentada, debido a que realiza automáticamente la umbralización, por ende, se obtiene un resultado más acertado.
* Se pueden presentar dificultades para la detección de figuras geométricas si no se tiene en cuenta la función cv2.findContours que OpenCv brinda, lo cual no da como resultado una imagen en óptimas condiciones.

# BIBLIOGRAFIA

[1] Grupo bcnvision. (2017). Sistemas de iluminación para aplicaciones de visión artificial. Recuperado de <https://www.bcnvision.es/blog-vision-artificial/iluminacion-visionartificial2/#:~:text=La%20fuente%20de%20luz%20puede,fluorescente%2C%20ultravioleta%2C%20entre%20otros>

[2] Parra, J.A. (2016). Segmentación por umbralización, método de otsu. *Universidad Nacional de Quilmes.* Recuperado de <https://es.slideshare.net/JorgeAntonioParraSerquen/segmentacin-por-umbralizacin-mtodo-de-otsu>

[3] Programador Clic. (2020). Principio e implementación del algoritmo OTSU (método Otsu: método de varianza máxima entre clases). Recuperado de <https://programmerclick.com/article/47001069656/>