



Visión Artificial

Carlos Andrés Sánchez Ríos

Departamento de Ingeniería Electromecánica y
Mecatrónica

Instituto Tecnológico Metropolitano

Somos Innovación Tecnológica con *Sentido Humano*



Alcaldía de Medellín

Visión Artificial

Agenda de clase

Segmentación

Segmentación

Operadores umbral

Umbral

Binario

Extensión

Escala de grises

Adaptativo



Institución
Universitaria
Reacreditada en Alta Calidad

SEGMENTACIÓN

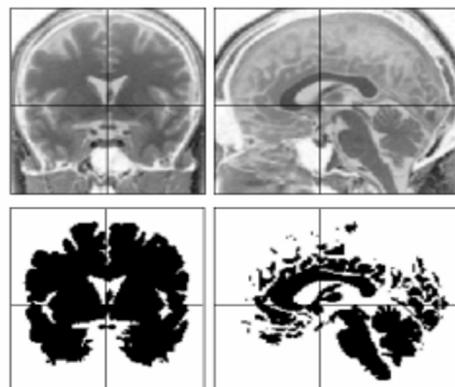
Somos Innovación Tecnológica con *Sentido Humano*



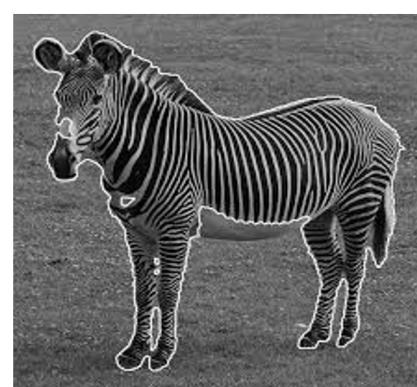
Alcaldía de Medellín

Segmentación

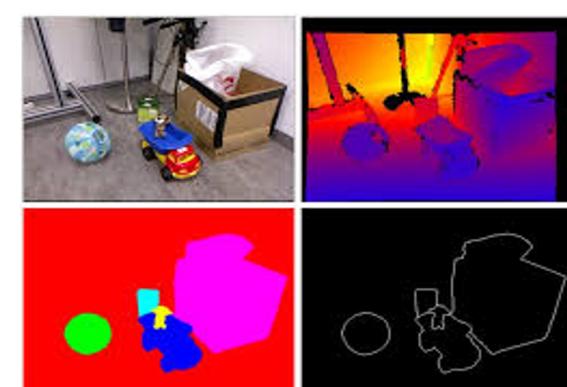
El objetivo de la segmentación es agrupar los píxeles por regiones de acuerdo a un criterio: Brillo, Color, Forma, Textura, y de acuerdo a la información usada para la separación se pueden dividir en técnicas basadas en **Similitud** o **Discontinuidad**.



Segmentación de tejidos
en imágenes MR



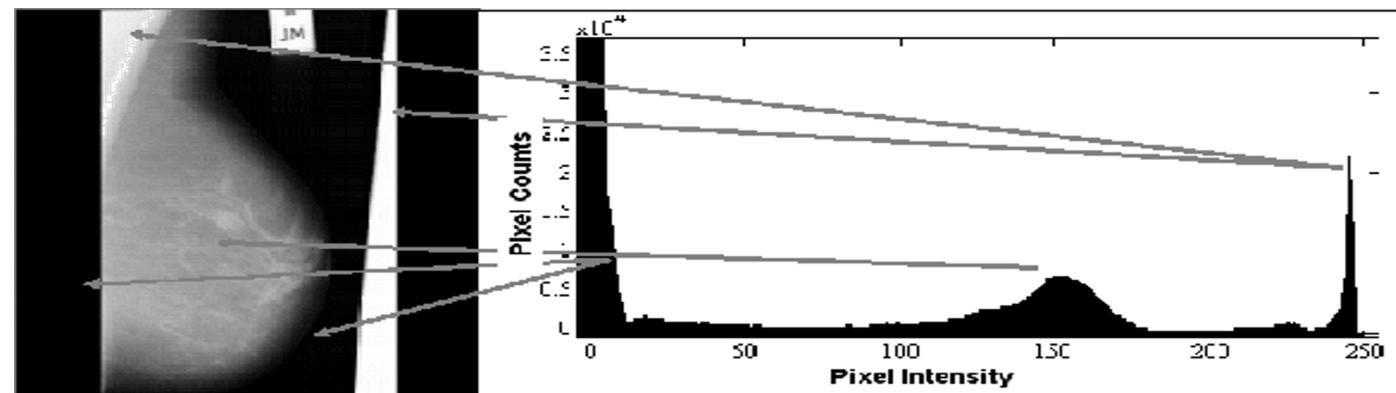
Segmentación por
contorno



Segmentación en
imágenes de rango

Segmentación

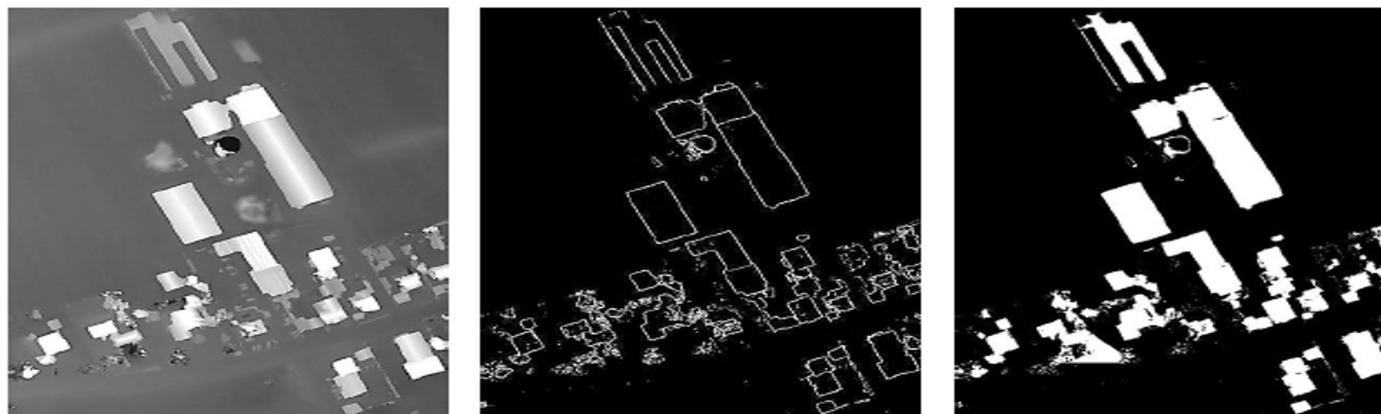
En la segmentación por **similitud** la imagen es particionada en regiones que son similares: por ejemplo los niveles de gris, en estos casos frecuentemente es usado el histograma para determinar el umbral adecuado.



Segmentación basada en el histograma

Segmentación

En la segmentación por **discontinuidad** se partitiona la imagen basada en cambios abruptos de intensidad, como los bordes en la imagen.



Segmentación basado en bordes



Institución
Universitaria
Reacreditada en Alta Calidad

OPERADORES UMBRAL

Somos Innovación Tecnológica con *Sentido Humano*

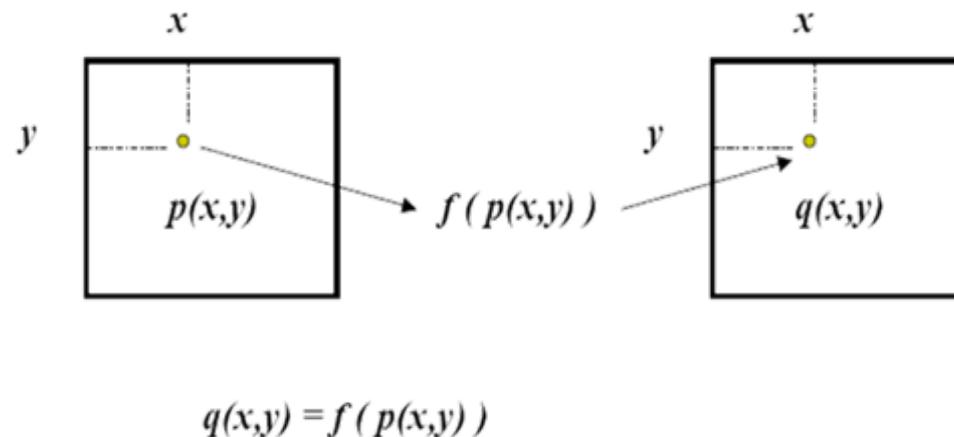


Alcaldía de Medellín

Segmentación

Operaciones Individuales

Se genera una nueva imagen de salida q a partir del procesamiento de cada píxel de una imagen de entrada p.

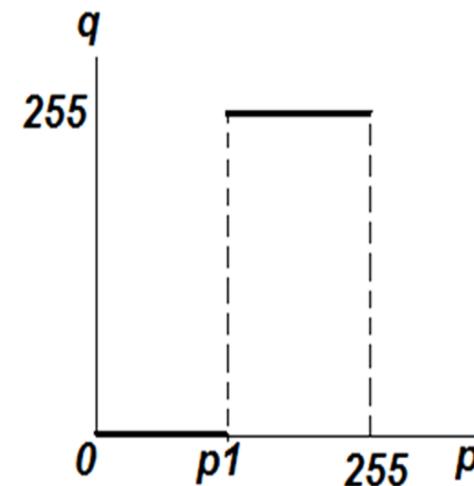


Segmentación

Operador Umbral

Crea una imagen de salida q con resolución en amplitud 1bpp (binarizada), a partir de un umbral.

$$q = \begin{cases} 0 & \text{para } p \leq p_1 \\ 255 & \text{para } p > p_1 \end{cases}$$

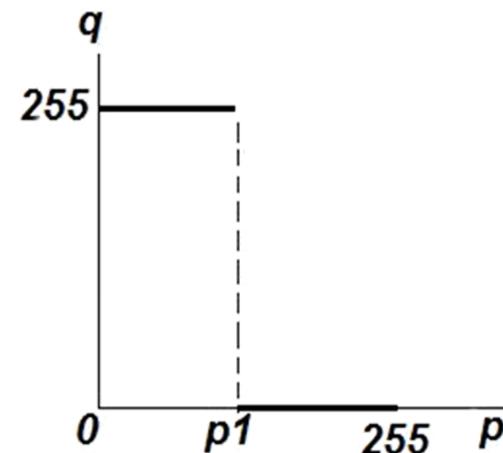


Segmentación

Operador Umbral Invertido

Crea una imagen de salida q con resolución en amplitud 1bpp (binarizada), a partir de un umbral.

$$q = \begin{cases} 255 & \text{para } p \leq p_1 \\ 0 & \text{para } p > p_1 \end{cases}$$

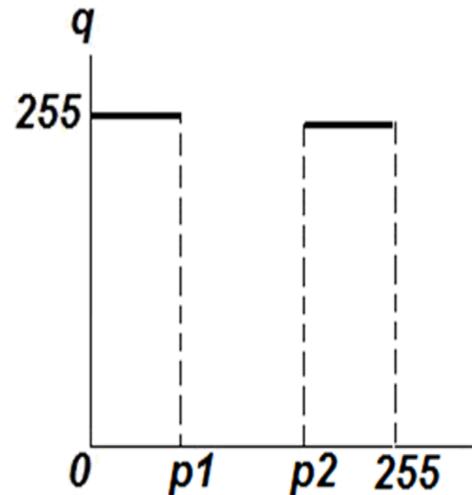


Segmentación

Operador Intervalo de Umbral Binario

Crea una imagen de salida q con resolución en amplitud 1bpp (binarizada), a partir de 2 umbrales p1 y p2.

$$q = \begin{cases} 255 & \text{para } p \leq p1 \text{ o } p \geq p2 \\ 0 & \text{para } p1 < p < p2 \end{cases}$$

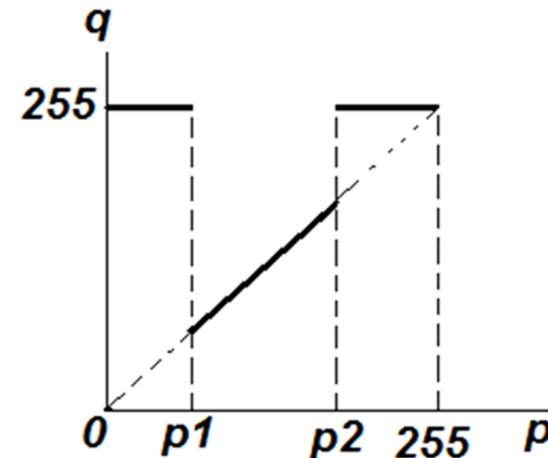


Segmentación

Operador de Umbral en Escala de Grises

Crea una imagen de salida q donde permanecen los pixeles con niveles de gris en el intervalo definido por p1 y p2.

$$q = \begin{cases} 255 & \text{para } p \leq p1 \text{ o } p \geq p2 \\ p & \text{para } p1 < p < p2 \end{cases}$$

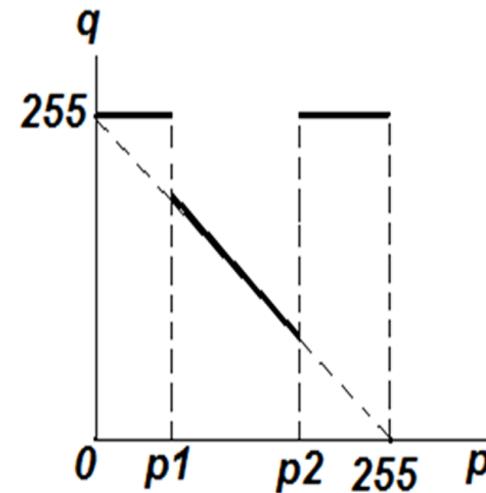


Segmentación

Operador de Umbral en Escala de Grises Invertido

Crea una imagen de salida q donde permanecen los pixeles con niveles de gris en el intervalo definido por p1 y p2.

$$q = \begin{cases} 255 & \text{para } p \leq p1 \text{ o } p \geq p2 \\ 255 - p & \text{para } p1 < p < p2 \end{cases}$$

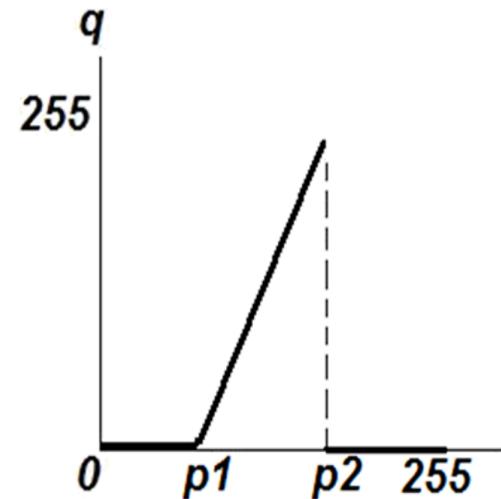


Segmentación

Operador de Extensión

Crea una imagen de salida q con la escala de grises completa entre el intervalo de entrada p1 y p2 y elimina los valores fuera del rango.

$$q = \begin{cases} 0 & \text{para } p \leq p1 \text{ o } p \geq p2 \\ (p - p1) \frac{255}{p2 - p1} & \text{para } p1 < p < p2 \end{cases}$$

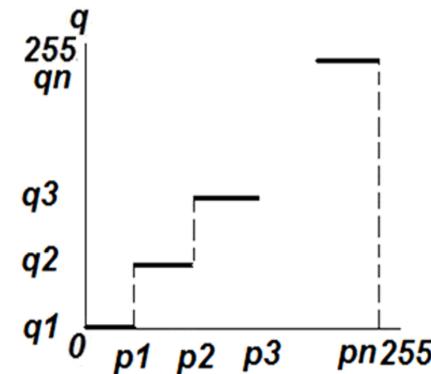


Segmentación

Operador de Reducción del Nivel de Gris

Crea una imagen de salida q con la escala de grises completa entre el intervalo de entrada p1 y p2 y elimina los valores fuera del rango.

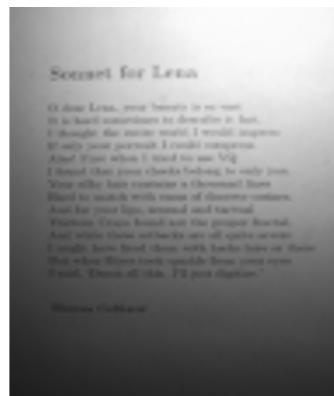
$$q = \begin{cases} 0 & \text{para } p \leq p_1 \\ q_1 & \text{para } p_1 < p \leq p_2 \\ q_2 & \text{para } p_2 < p \leq p_3 \\ \vdots & \vdots \\ q_n & \text{para } p_{n-1} < p \leq 255 \end{cases}$$



Segmentación

Umbrales Adaptativos

Los operadores umbrales vistos anteriormente usan un umbral global para todos los pixeles de la imagen, lo que en muchos casos no es adecuado, teniendo en cuenta que las condiciones de iluminación sobre la escena pueden variar. En estos casos lo más adecuado es utilizar umbrales adaptativos, que cambien dinámicamente de acuerdo a los niveles de intensidad de cada pixel y su vecindario.



Imágenes tomadas de <http://homepages.inf.ed.ac.uk/rbf/HIPR2/adptrsh.htm>

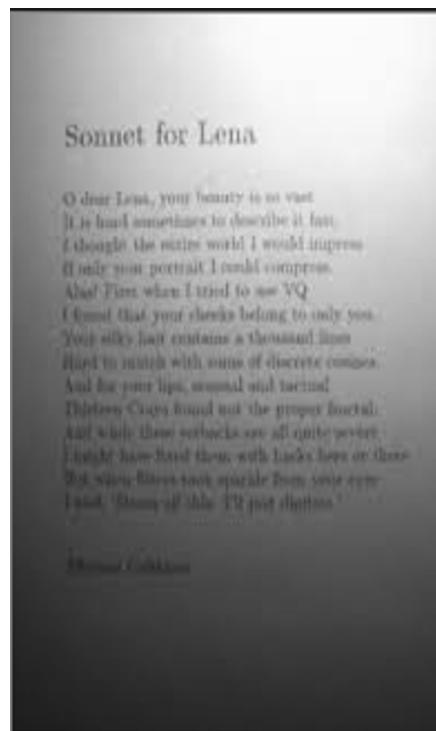


Imagen Original

Sonnet for Lena

O dear Lena, your beauty is so vast,
It is hard sometimes to describe it; but
I thought the entire world I could impress
(Only your portrait I could compose).
Also! First when I tried to see you
(I found that your cheeks belong to only you)
Your silky hair contains a thousand lines
Hard to match with some of discrete curves
And for your lips, smooth and tamed
Thirteen. They had not the proper fount.
And while those wrinkles are all quite severe
I might have fixed them with hales here or there
But when I tried to make from your eyes
Just "Ghosts" of this I did not succeed.

...
...
...

Sonnet for Lena

O dear Lena, your beauty is so vast,
It is hard sometimes to describe it; but
I thought the entire world I could impress
(Only your portrait I could compose).
Also! First when I tried to see you
(I found that your cheeks belong to only you)
Your silky hair contains a thousand lines
Hard to match with some of discrete curves
And for your lips, smooth and tamed
Thirteen. They had not the proper fount.
And while those wrinkles are all quite severe
I might have fixed them with hales here or there
But when I tried to make from your eyes
Just "Ghosts" of this I did not succeed.

...
...
...

Umbral Adaptativo

T= Gaussiano

Imágenes tomadas de <http://homepages.inf.ed.ac.uk/rbf/HIPR2/adptrsh.htm>



Institución
Universitaria
Reacreditada en Alta Calidad

OPENCV

Somos Innovación Tecnológica con *Sentido Humano*



Alcaldía de Medellín

Función de OpenCV

`cv2.threshold(src, thresh, maxval, type[, dst])`

1. src - imagen fuente en escala de grises.
2. thresh - valor de umbral.
3. maxval - es el valor máximo a usar por las funciones de umbralización, representa el valor que tomarán los pixeles al ser mayores o menores al valor de umbralización.
4. type - método de umbralización
`cv2.THRESH_BINARY`
`cv2.THRESH_BINARY_INV` `cv2.THRESH_TRUNC`
`cv2.THRESH_TOZERO` `cv2.THRESH_TOZERO_INV`

- **THRESH_BINARY**

$$dst(x,y) = \begin{cases} \text{maxval} & \text{if } \text{src}(x,y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$$

- **THRESH_BINARY_INV**

$$dst(x,y) = \begin{cases} 0 & \text{if } \text{src}(x,y) > \text{thresh} \\ \text{maxval} & \text{otherwise} \end{cases}$$

- **THRESH_TRUNC**

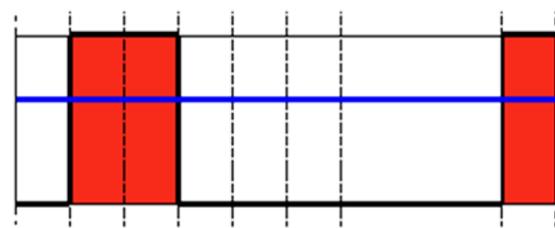
$$dst(x,y) = \begin{cases} \text{threshold} & \text{if } \text{src}(x,y) > \text{thresh} \\ \text{src}(x,y) & \text{otherwise} \end{cases}$$

- **THRESH_TOZERO**

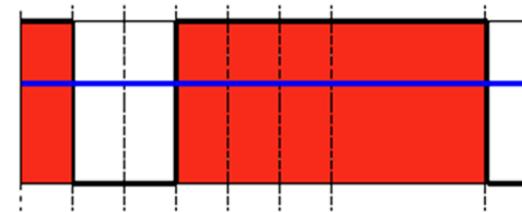
$$dst(x,y) = \begin{cases} \text{src}(x,y) & \text{if } \text{src}(x,y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$$

- **THRESH_TOZERO_INV**

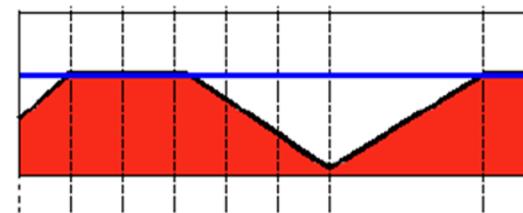
$$dst(x,y) = \begin{cases} 0 & \text{if } \text{src}(x,y) > \text{thresh} \\ \text{src}(x,y) & \text{otherwise} \end{cases}$$



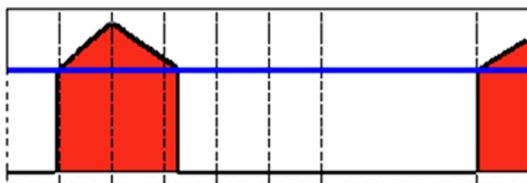
Binario



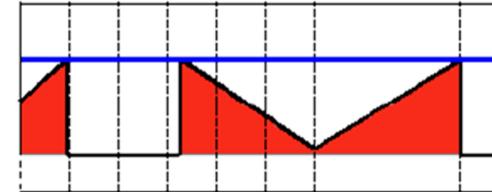
Binario inverso



Truncado



Cero



Cero inverso

Imagen tomada de <http://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html>



Institución
Universitaria
Reacreditada en Alta Calidad

EJERCICIOS

Somos Innovación Tecnológica con *Sentido Humano*



Alcaldía de Medellín

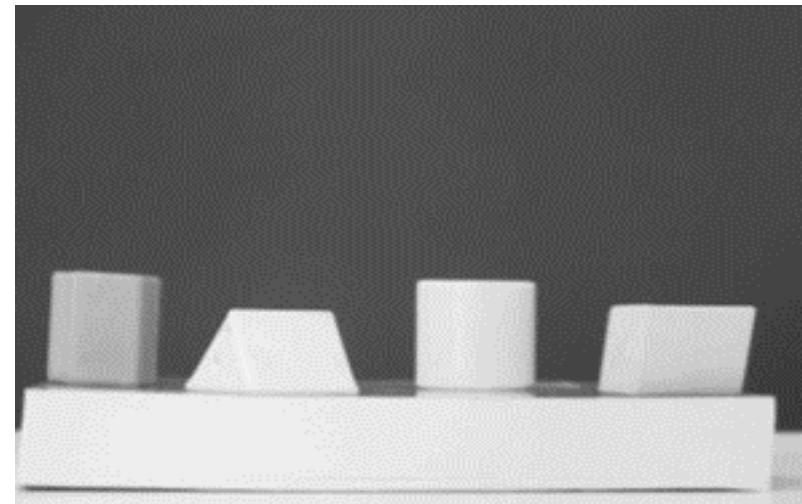
Algoritmo para escoger umbral global automático

1. Seleccione un umbral inicial T . Podría ser el valor promedio de la imagen.
2. Segmente la imagen usando T , lo cual producira 2 grupos de pixeles $G1$ para todos los pixeles con niveles de intensidad mayores a T y $G2$ para los menores a T .
3. Calcule los valores de niveles de intensidad de gris promedios $mean1, mean2$ para las regions $G1$ y $G2$.
4. Calcular un nuevo umbral
$$T = (1/2)(mean1 + mean2)$$
5. Repetir los pasos 2 al 4 hasta que la diferencia en T en iteraciones sucesivas sea menor que un parametro e predefinido.

Implementación de Operadores en OpenCV

EJERCICIO #1:

Realice un programa que calcule el área en píxeles del objeto mas a la izquierda en la figura.



Implementación de Operadores en OpenCV

EJERCICIO #2:

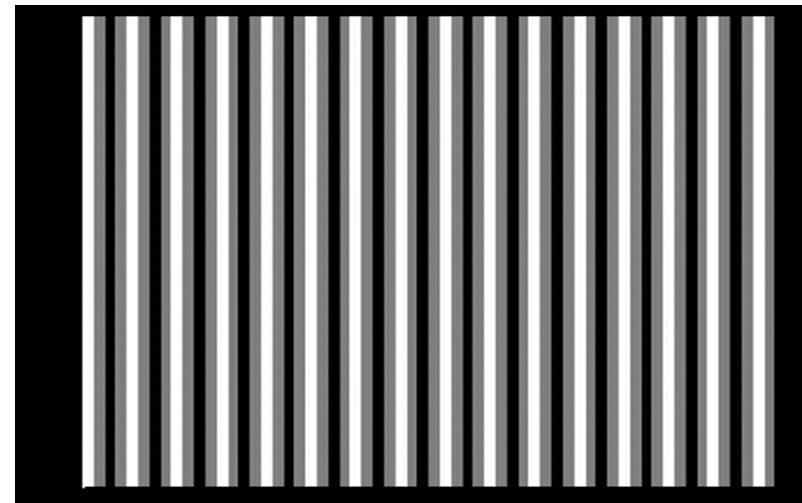
Realice un programa para detectar la suciedad en la botella mostrada en la figura.



Implementación de Operadores en OpenCV

EJERCICIO #3:

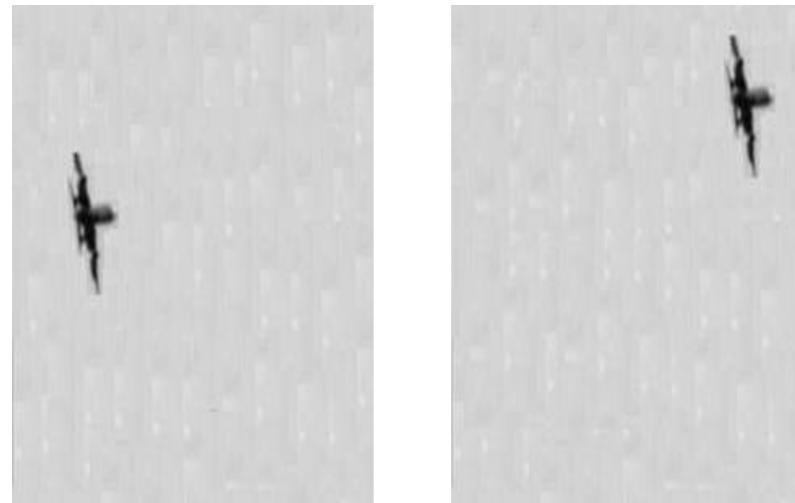
Realice un programa que calcule el Número de barras blancas en la imagen y la separación entre ellas, suponiendo que todas las adyacentes están a una misma distancia.



Implementación de Operadores en OpenCV

EJERCICIO #4:

Realice un programa que halle la velocidad en píxeles/segundo que lleva el avión, si la cámara de video toma 30 frames por segundo.





¡Gracias!

Somos Innovación Tecnológica con *Sentido Humano*



Alcaldía de Medellín