



Instituto Politécnico Nacional.



Escuela Superior de Cómputo.

## Análisis de Imágenes

Segmentación y ajuste de brillo

Integrantes:

Frías Mercado Carlos Elliot – 2016630119.

Grupo:  
3CM5

Profesora:  
Cruz Meza María Elena

# Contenido

Introducción .....	4
Objetivo .....	4
Marco Teórico .....	4
Ajuste de brillo .....	7
Desplazamiento del histograma.....	8
Contracción del histograma .....	9
Expansión del histograma .....	10
Ecualización.....	11
Segmentación parcial.....	14
Convolución.....	14
Filtros pasa altas.....	15
Filtro de Prewitt.....	15
Filtro de Roberts.....	16
Filtro de Sobel .....	16
Filtro Laplaciano .....	17
Filtro diferencia de píxeles .....	19
Filtro diferencia de píxeles separados.....	19
Filtros pasa bajas .....	20
Filtro promedio .....	20
Filtro promedio pesado.....	21
Filtro gaussiano .....	22
Filtros no lineales .....	23
Filtro media .....	23
Filtro mediana .....	24
Filtro moda .....	24
Filtro máximo .....	25
Filtro mínimo .....	25
Segmentación global .....	26
Binarización .....	27
Binarización de Otsu.....	28
Umbralización .....	31
Segmentación de una imagen.....	32

Conclusiones .....	37
Referencias.....	38

## Introducción

El análisis de imágenes es una herramienta muy poderosa ya que nos permite conocer a detalle una imagen y sus componentes, incluso partes de una imagen que no podemos ver a simple vista, dentro del gran campo que es el análisis de imágenes se encuentra la segmentación la cual es de sus herramientas más importantes ya que nos permite tomar una imagen y desechar aquello que no nos sirve para analizar únicamente alguna parte que nos interese, o separar los componentes de la imagen y analizarlos individualmente. Aquí podremos conocer más a detalle la segmentación de imágenes y a su vez un poco del ajuste del brillo, que igual nos permite mejorar una imagen para obtener una mejor segmentación, además de que se explicarán los filtros y métodos de segmentación usados en las imágenes y el resultado y diferencias que cada uno de ellos produce.

## Objetivo

Aplicar las técnicas de análisis de imágenes estudiadas en clase para la mejora de la imagen, eliminación de ruido, segmentación parcial y completa, con el fin de demostrar mis conocimientos, habilidades y competencias adquiridos durante el curso al resolver un problema real.

## Marco Teórico

El objetivo de la segmentación de imágenes es agrupar los píxeles en regiones de imagen salientes, es decir, regiones correspondientes a superficies individuales, objetos o partes naturales de objetos. Se puede usar la segmentación para el reconocimiento de objetos, la estimación de límites de oclusión dentro de los sistemas de movimiento, la compresión de imágenes, la edición de imágenes o la búsqueda en la base de datos de imágenes.

Para la entrada, se considera principalmente el brillo de la imagen, aunque se pueden usar técnicas similares con información de color, movimiento y / o disparidad estéreo <sup>[1]</sup>

La segmentación divide una imagen en distintas regiones que contienen píxeles con atributos similares. Para ser significativo y útil para el análisis e interpretación de imágenes, las regiones deben relacionarse fuertemente con objetos representados o características de interés. La segmentación significativa es el primer paso desde el procesamiento de imágenes de bajo nivel que transforma una imagen en escala de grises o color en una o más imágenes a una descripción de alto nivel en términos de características, objetos y escenas.

Las técnicas de segmentación pueden ser contextuales o no contextuales. Los últimos no tienen en cuenta las relaciones espaciales entre las entidades en una imagen y agrupan los píxeles en función de algún atributo global, por ejemplo, el nivel de gris o color. Las técnicas contextuales además explotan estas relaciones, por ejemplo, agrupando los píxeles con niveles de gris similares y cerrando las ubicaciones espaciales.

## Segmentación no contextual

El umbralización o binarización es la técnica de segmentación no contextual más simple. Con un solo umbral, transforma una imagen en escala de grises o color en una imagen binaria considerada como un mapa de región binaria, el mapa binario contiene dos regiones posiblemente disjuntas, una de ellas con píxeles con valores de datos de entrada más pequeños que un umbral y otra relacionada con los valores de entrada que están en o por encima del umbral. Las regiones primera y última se etiquetan generalmente con etiquetas cero (0) y uno (1), respectivamente. La segmentación depende de la propiedad de la imagen que se está umbralizando y de cómo se elige el umbral.

La propiedad de imagen más común para la umbralización es por el nivel de gris de píxel:

$$g(x, y) = 0 \text{ si } f(x, y) < T$$

$$g(x, y) = 1 \text{ si } f(x, y) \geq T$$

Donde T es el umbral

Usando dos umbrales,  $T_1 < T_2$ , se puede definir un rango de niveles de gris relacionados con la región:

$$g(x, y) = 0 \text{ si } f(x, y) < T_1 \text{ O } f(x, y) > T_2$$

$$g(x, y) = 1 \text{ si } T_1 \leq f(x, y) \leq T_2.$$



Imagen 1: Binarización de una imagen con umbral  $U = 94$

En general, los umbrales no contextuales pueden incluir dos o más umbrales, así como producir más de dos tipos de regiones, de modo que los rangos de las señales de imagen de entrada relacionadas con cada tipo de región se separan con todos éstos umbrales. La

cuestión del umbral múltiple o multi umbralización es cómo determinar automáticamente el valor de cada uno de los umbrales.



Imagen 2: Segmentación de una imagen con múltiples umbrales (10, 20, 50, 100, 200)

### Segmentación contextual

Los umbrales no contextuales agrupan los píxeles sin tener en cuenta sus ubicaciones relativas en el plano de la imagen. La segmentación contextual puede ser más exitosa en la separación de objetos individuales, ya que da cuenta de la proximidad de los píxeles que pertenecen a un objeto individual.

Dos enfoques básicos para la segmentación contextual se basan en la discontinuidad o similitud de la señal. Las técnicas basadas en la discontinuidad intentan encontrar límites completos que encierran regiones relativamente uniformes suponiendo cambios abruptos en la señal a través de cada límite. Las técnicas basadas en similitudes intentan crear directamente estas regiones uniformes agrupando los píxeles conectados que satisfacen ciertos criterios de similitud. Ambos enfoques se reflejan entre sí, en el sentido de que un límite completo divide una región en dos. <sup>[2]</sup>

Entre los ejemplos de la segmentación contextual se encuentran:

- Conectividad de pixeles
- Similitud de regiones
- Crecimiento de regiones
- Segmentación por el método divide y une

## Ajuste de brillo

El ajuste de brillo nos permite manipular directamente el histograma de una imagen para ajustar su nivel de brillo, ya sea aclarando la imagen oscureciéndola, o equilibrando los niveles de brillo en el histograma para que no se concentren en una sola zona.

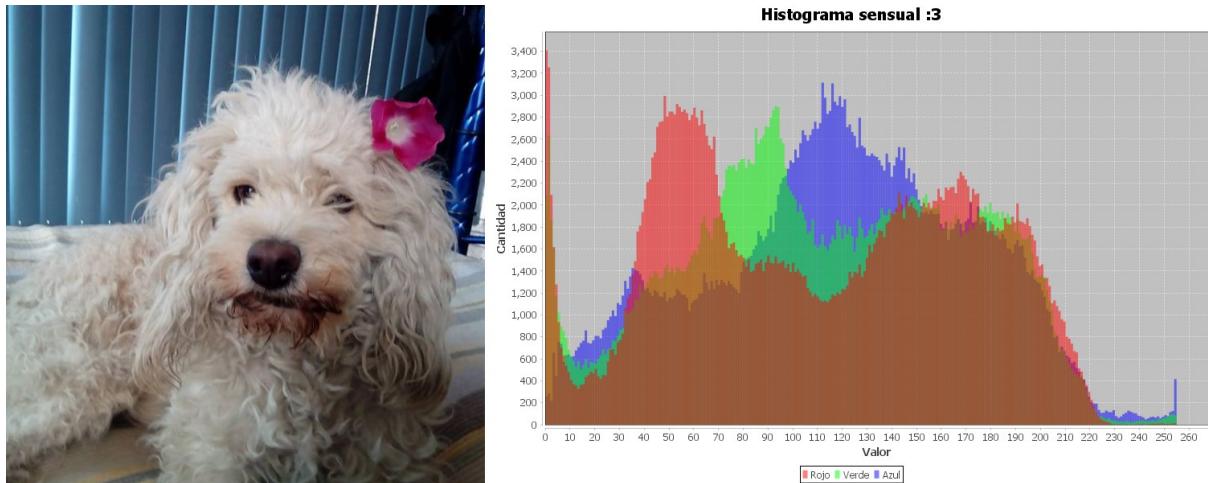


Imagen 3: Histograma de una imagen

Estos ajustes buscan mejorar la calidad de una imagen a través de la modificación de su histograma para que ésta no sea ni muy clara ni muy oscura, pudiendo apreciar mejor cada detalle de la misma.

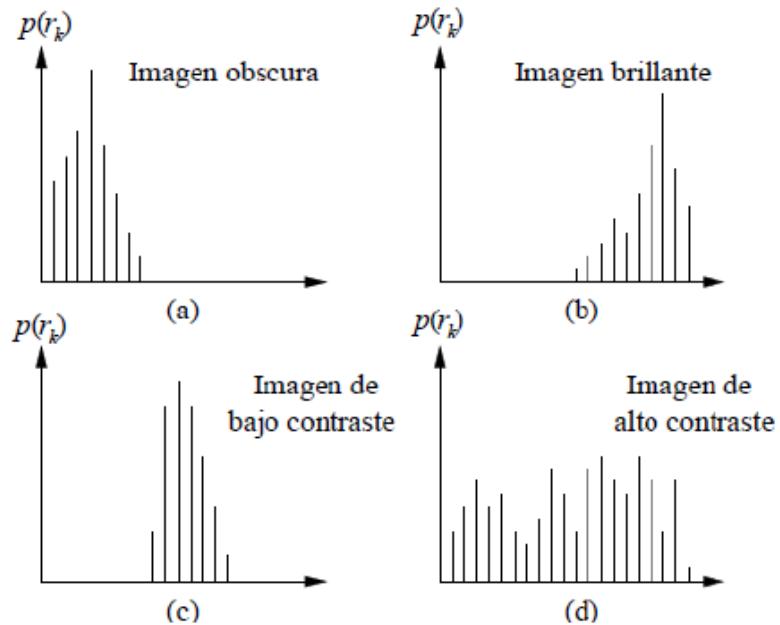


Imagen 4: Diferentes tipos de histogramas en las imágenes

## Desplazamiento del histograma

El desplazamiento del histograma consiste en agregar un valor constante, ya sea positivo o negativo a todos los niveles de brillo del histograma con el fin de moverlo o desplazarlo hacia los niveles más claros o a los más oscuros.

Está expresado de la forma:

$$G(x,y) = F(x,y) + DES$$

Donde:

$F(x,y)$  es la imagen original

$DES$  es una constante de desplazamiento la cual puede ser positiva o negativa

$G(x,y)$  es la nueva imagen



Imagen 5: Desplazamiento del histograma de una imagen con un  $DES = 50$

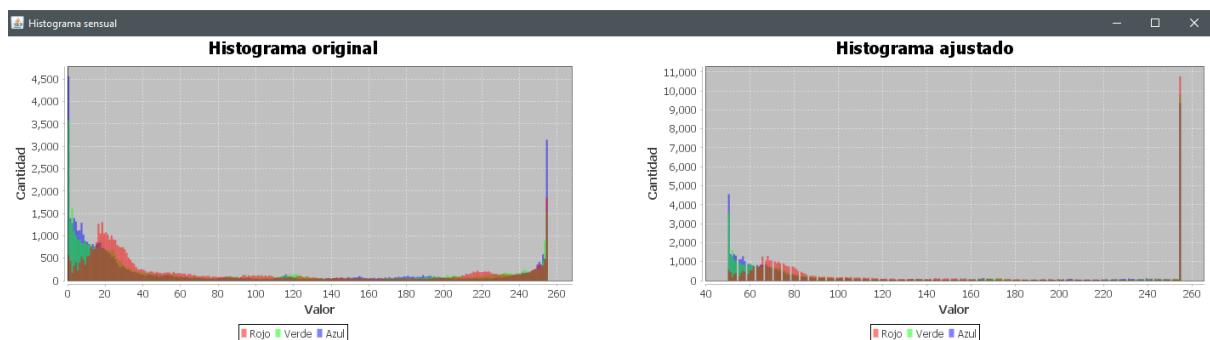


Imagen 6: Histograma de una imagen desplazado con un  $DES = 50$

## Contracción del histograma

La contracción consiste en limitar los valores de brillo máximos y mínimo del histograma, haciendo que ningún valor del mismo sobrepase dichos límites, se expresa mediante:

$$G(x,y) = \left( \frac{C_{\max} - C_{\min}}{F(x,y)_{\max} - F(x,y)_{\min}} \right) * (F(x,y) - F(x,y)_{\min}) + C_{\min}$$

Donde:

$G(x,y)$  es la nueva imagen.

$C_{\max}$  es el valor de brillo máximo deseado en el histograma.

$C_{\min}$  es el valor de brillo mínimo deseado en el histograma.

$F(x,y)_{\max}$  es el nivel de brillo más alto en la imagen de entrada.

$F(x,y)_{\min}$  es el nivel de brillo más bajo en la imagen de entrada.

$F(x,y)$  es la imagen de entrada



Imagen 7: Contracción del histograma de una imagen en el rango de brillo de 100 – 200

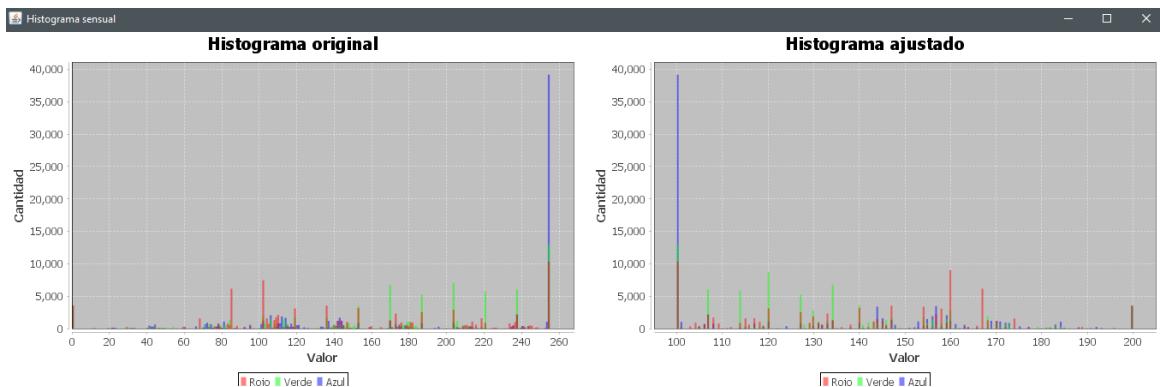


Imagen 8: Histograma contraído dentro de rango de nivel de brillo 100 a 200

## Expansión del histograma

Consiste en extender los niveles de brillo de la imagen de tal forma que éstos se encuentren distribuidos sobre todos los niveles del histograma, evitando que se encuentren agrupados en una misma zona del histograma. Se expresa mediante la fórmula:

$$G(x,y) = \left( \frac{F(x,y) - F(x,y)\min}{F(x,y)\max - F(x,y)\min} \right) * (n\text{Max} - n\text{Min}) + n\text{Min}$$

Donde:

$G(x,y)$  es la nueva imagen.

$n\text{Max}$  es el valor de brillo máximo posible (255 para imágenes de 8 bits)

$n\text{Min}$  es el valor de brillo mínimo posible (255 para imágenes de 8 bits)

$F(x,y)\max$  es el nivel de brillo más alto en la imagen de entrada.

$F(x,y)\min$  es el nivel de brillo más bajo en la imagen de entrada.

$F(x,y)$  es la imagen de entrada



Imagen 9: Mejora de una imagen mediante la expansión de su histograma.

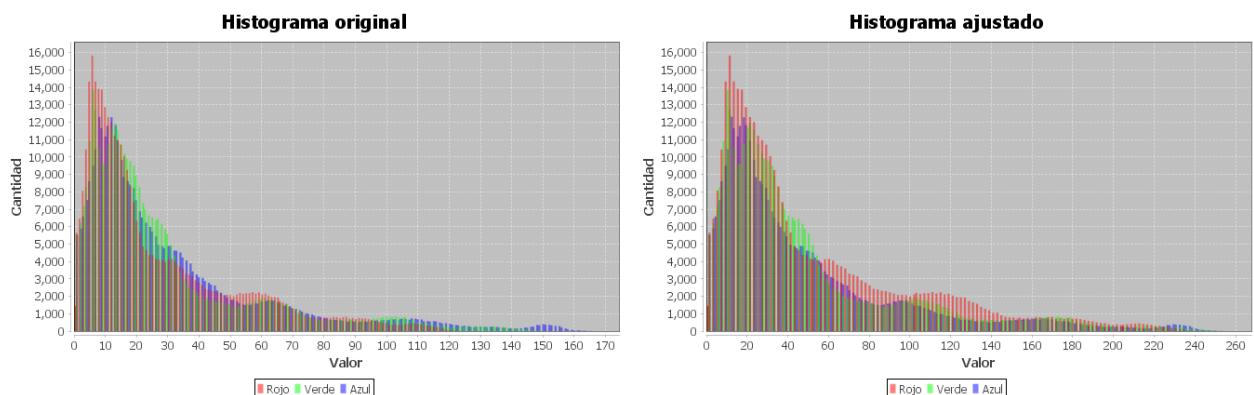


Imagen 10: Histograma de una imagen contra el histograma expandido de la misma imagen.

## Ecualización

La ecualización del histograma es un método de procesamiento imágenes que busca ajustar el contraste de una imagen modificando la distribución de la intensidad del histograma. El objetivo de esta técnica es dar una tendencia lineal a la función de probabilidad acumulada asociada a la imagen.

El procesamiento de la ecualización de histogramas se basa en el uso de la función de probabilidad acumulada (cdf). El cdf es una suma acumulativa de todas las probabilidades que se encuentran en su dominio. La idea de este procesamiento es otorgar a la imagen resultante una función de distribución acumulativa lineal.

Se calcula mediante la fórmula:

$$S_k = T(r_k) = (L - 1) \sum_{j=0}^k Pr(r_j) = \frac{(L - 1)}{MN} \sum_{j=0}^k n_j$$

Donde:

$S_k$  = Nuevo valor de brillo que tomarán todos los pixeles  $n_k$

$r_k$  = Nivel de brillo k

$T(r_k)$  = Transformación del pixel con nivel de brillo k

L = Máximo nivel de brillo posible (255 para imágenes de 8 bits)

$Pr(r_j)$  = Probabilidad de encontrar un pixel con valor de brillo j

M, N = Largo y ancho de la imagen

$n_j$  = acumulado de pixeles con nivel de brillo j

## Ejemplo de ecualización

Una imagen de 3 bits y de tamaño 64 x 64 pixeles, tiene la siguiente distribución de intensidades:

$r_k$	Nivel de brillo	$n_k$
$r_0$	0	790
$r_1$	1	1023
$r_2$	2	850
$r_3$	3	656
$r_4$	4	329
$r_5$	5	245
$r_6$	6	122
$r_7$	7	81

Tabla 1: Distribución de intensidades de la imagen de 3 bits ejemplificada.

Recordando que  $n_k$  es la cantidad de pixeles dentro de la imagen que tienen el valor de brillo  $r_k$ , la frecuencia de aparición de los mismos, sabiendo que  $r_k$  va de  $r_k = 0$  hasta  $r_k = 7$  por tratarse de una imagen de 3 bits, para una imagen de 8 bits  $r_k$  correría de  $r_k = 0$  hasta  $r_k = 255$ .

Al calcular la probabilidad de encontrar cada nivel de brillo dentro de la imagen  $Pr(r_k)$ , en donde  $Pr(r_k) = n_k / MN$  y  $M = 64$ ,  $N = 64$ , se obtiene la siguiente tabla:

$r_k$	Nivel de brillo	$n_k$	$n_k / MN$	$Pr(r_k)$
$r_0$	0	790	790/4096	0.19
$r_1$	1	1023	1023/4096	0.25
$r_2$	2	850	850/4096	0.21
$r_3$	3	656	656/4096	0.16
$r_4$	4	329	329/4096	0.08
$r_5$	5	245	245/4096	0.06
$r_6$	6	122	122/4096	0.03
$r_7$	7	81	81/4096	0.02

Tabla 2: Tabla extendida con los valores de frecuencia y probabilidad calculados

Para terminar, se procede al cálculo de la sumatoria:

$$S_k = (L - 1) \sum_{j=0}^k Pr(r_k)$$

Para  $r_k = 0$  con  $L = 8$ :

$$S_k = (8 - 1) * 0.19 = 7 * 0.19 = 1.33$$

Para  $r_k = 1$  con  $L = 8$ :

$$S_k = (8 - 1) * (0.19 + 0.25) = 7 * 0.44 = 3.08$$

Para  $r_k = 10$  con  $L = 8$ :

$$S_k = (8 - 1) * (0.19 + 0.25 + 0.21) = 7 * 0.65 = 4.55$$

$r_k$	Nivel de brillo	$n_k$	$n_k / MN$	$Pr(r_k)$	$S_k$	Nuevo nivel de brillo
$r_0$	0	790	790/4096	0.19	1.33	1
$r_1$	1	1023	1023/4096	0.25	3.08	3
$r_2$	2	850	850/4096	0.21	4.55	5
$r_3$	3	656	656/4096	0.16	5.67	6
$r_4$	4	329	329/4096	0.08	6.23	6
$r_5$	5	245	245/4096	0.06	6.65	7
$r_6$	6	122	122/4096	0.03	6.86	7
$r_7$	7	81	81/4096	0.02	7.00	7

Tabla 3: Tabla completa de la ecualización de la imagen

Una vez terminada la tabla de ecualización de la imagen y habiendo calculado todos los  $S_k$  se ajusta o redondea el valor para obtener un número entero, el cuál será el nuevo nivel de brillo, aplicando esto se cambiará el valor  $r_k$  por el del  $S_k$  para todos los pixeles  $n_k$ .

Aplicando al ejemplo, los 790 pixeles  $n_0$  y cuyo nivel de brillo es  $r_k = 0$ , tendrán un nuevo nivel de brillo el cual será  $s_k = 1$ .



Imagen 11: Imagen original junto a la misma imagen tras haber sido ecualizada.

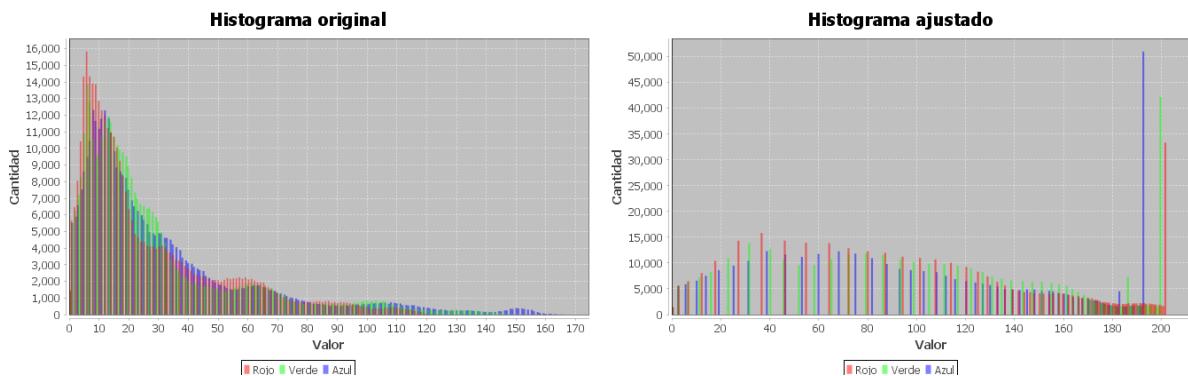


Imagen 12: Histograma original junto a la ecualización del mismo.

## Segmentación parcial

El filtrado es una técnica para modificar o mejorar una imagen, puede filtrar una imagen para enfatizar ciertas funciones o eliminar otras. Las operaciones de procesamiento de imágenes implementadas con filtrado incluyen suavizado, nitidez y mejora de bordes. El filtrado es una operación sobre la vecindad de píxeles, en la cual el valor de cualquier píxel dado en la imagen de salida se determina aplicando algún algoritmo a los valores de los píxeles en la vecindad del píxel de entrada correspondiente. La vecindad de un píxel es un conjunto de píxeles, definidos por sus ubicaciones en relación con ese píxel. El filtrado lineal es aquel en el que el valor de un píxel de salida es una combinación lineal de los valores de los píxeles en el vecindario del píxel de entrada.<sup>[3]</sup>

## Convolución

El filtrado lineal de una imagen se realiza a través de una operación llamada convolución. La convolución es una operación de entorno en la que cada píxel de salida es la suma ponderada de los píxeles de entrada vecinos. La matriz de pesos se denomina núcleo de convolución, también conocido como filtro. Un kernel de convolución es un kernel de correlación que se ha girado 180 grados.

Pensando en una imagen como la siguiente y con un kernel:

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

8	1	6
3	5	7
4	9	2

Figura 1: Matriz de valores de la imagen A de 5 x 5 y un kernel T de 3x3

Para calcular el píxel de salida en la posición (3,1) de la imagen A:

1. Gire el kernel 180 grados sobre su elemento central para crear un kernel de convolución.
2. Deslice el elemento central del kernel de convolución para que quede sobre el elemento (3,1) de A.
3. Multiplique cada valor en el kernel de convolución por el píxel de A debajo.
4. Sume los productos individuales del paso anterior.

Por lo tanto, el píxel de salida (3,1) es

$$B(3,1) = (1 * 2) + (8 * 9) + (15 * 4) + (7 * 7) + (14 * 5) + (16 * 3) + (13 * 6) + (20 * 1) + (22 * 8)$$

$$B(3,1) = 585$$

## Filtros pasa altas

Se puede usar un filtro de pasa altas para hacer que una imagen se vea más nítida. Estos filtros enfatizan detalles finos en la imagen, exactamente lo opuesto al filtro de paso bajo. El filtrado de pasa altas funciona exactamente de la misma manera que el filtrado pasa bajas; solo usa un kernel de convolución diferente.

Mientras que los filtros pasa bajas suavizan el ruido, el filtrado de pasa altas hace exactamente lo contrario: amplifica el ruido. Puede salirse con la suya si la imagen original no es demasiado ruidosa; De lo contrario el ruido abrumará la imagen.

Los filtros pasa altas también pueden hacer que los detalles pequeños y débiles se exageren enormemente. Una imagen sobre procesada se verá granulada y poco natural, y las fuentes puntuales tendrán donas oscuras a su alrededor. Así, mientras que el uso de filtros pasa altas puede mejorar una imagen al agudizar los detalles, la exageración puede degradar significativamente la calidad de la misma.

Un filtro pasa alto es un filtro que pasa bien las frecuencias altas, pero atenúa las frecuencias menores a la frecuencia de corte, el filtrado es fundamentalmente una operación en el dominio de la frecuencia. [4][5]

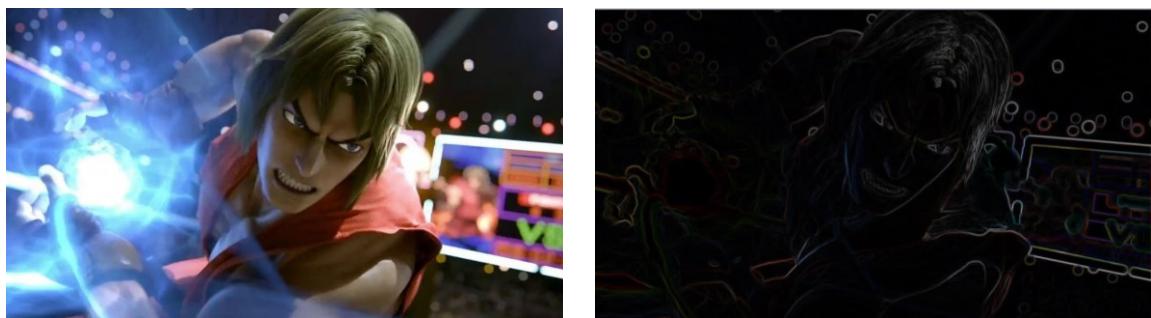


Imagen 13: Imagen original junto a la misma tras haberse aplicado un filtro pasa altas (diferencia de pixeles)

## Filtro de Prewitt

El filtro de Prewitt es un operador basado en la primera derivada de la función Gaussiana, es usado para la detección de bordes. Utiliza dos kernel diferentes donde uno es usado para el eje horizontal mientras que otro es para el eje vertical, además de contar con un factor de balance, una constante K que se multiplica por cada kernel y tiene un valor de  $k = 1/3$

Está representado por los siguientes Kernel:

$$Gx = \begin{array}{|c|c|c|} \hline +1 & 0 & -1 \\ \hline +1 & 0 & -1 \\ \hline +1 & 0 & -1 \\ \hline \end{array} \quad Gy = \begin{array}{|c|c|c|} \hline +1 & +1 & +1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Figura 2: Kernel de convolución para filas y columnas del filtro de Prewitt.



Imagen 14: Imagen tras haberse aplicado el filtro de Prewitt.

### Filtro de Roberts

El filtro de Robert es un operador basado en la primera derivada de la función Gaussiana, es usado para la detección de bordes. Utiliza dos kernel diferentes donde uno es usado para el eje horizontal mientras que otro es para el eje vertical.

Está representado por los siguientes Kernel:

$$G_x = \begin{array}{|c|c|c|} \hline 1 & 0 & 0 \\ \hline 0 & -1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad G_y = \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline -1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

Figura 3: Kernel de convolución para filas y columnas del filtro de Roberts.



Imagen 15: Imagen tras haberse aplicado el filtro de Roberts.

### Filtro de Sobel

El filtro de Solbel es un operador basado en la primera derivada de la función Gaussiana, es usado para la detección de bordes. Utiliza dos kernel diferentes donde uno es usado para el eje horizontal mientras que otro es para el eje vertical, además de contar con un factor de balance, una constante K que se multiplica por cada kernel y tiene un valor de  $k = \frac{1}{4}$

Está representado por los siguientes Kernel:

$$Gx = \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \quad Gy = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$$

Figura 4: Kernel de convolución para filas y columnas del filtro de Sobel.



Imagen 16: Imagen tras haberse aplicado el filtro de Sobel.

### Filtro Laplaciano

Los filtros laplaciados son filtros derivados que se usan para encontrar áreas de cambio rápido (bordes) en las imágenes. Dado que los filtros derivados son muy sensibles al ruido, es común suavizar la imagen antes de aplicar el Laplaciano. Este proceso de dos pasos se denomina operación de Laplaciano de Gauss (LoG), el operador de LoG toma la segunda derivada de la imagen, si la imagen es uniforme, el LoG dará cero, pero donde ocurra un cambio, el LoG dará una respuesta positiva en el lado más oscuro y una respuesta negativa en el lado más claro.<sup>[6]</sup>

En un borde afilado entre dos regiones, la respuesta será:

- Cero lejos del borde
- Positivo solo a un lado
- Negativo solo al otro lado
- Cero en algún punto intermedio en el borde mismo

Está representado por los siguientes Kernel:

$$A = \begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} \quad B = \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$
  

$$C = \begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} \quad D = \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 9 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Figura 5: Kernel de convolución de los filtros Laplaciano.

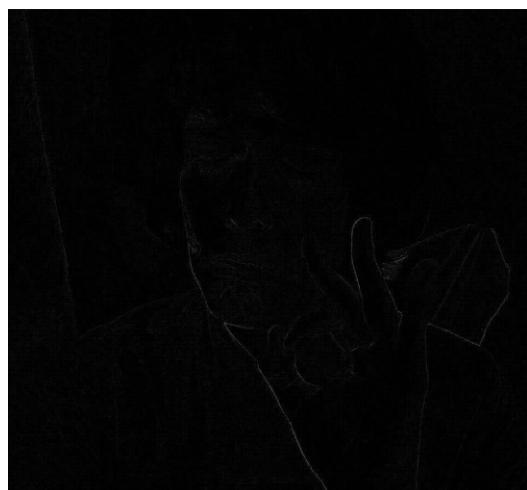


Imagen 17: Foto original con su equivalente tras aplicar el Laplaciano tipo A



Imagen 18: Foto original con su equivalente tras aplicar el Laplaciano tipo B

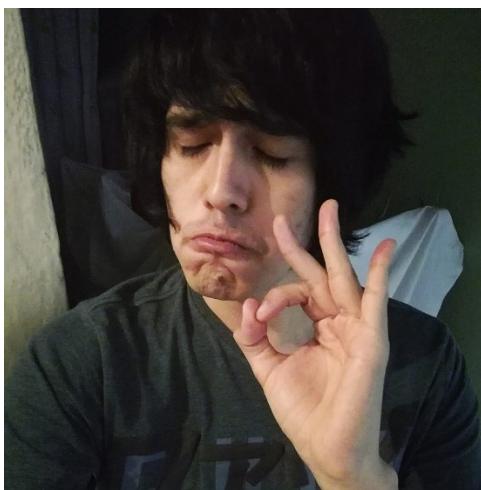


Imagen 19: Foto original con su equivalente tras aplicar el Laplaciano tipo C

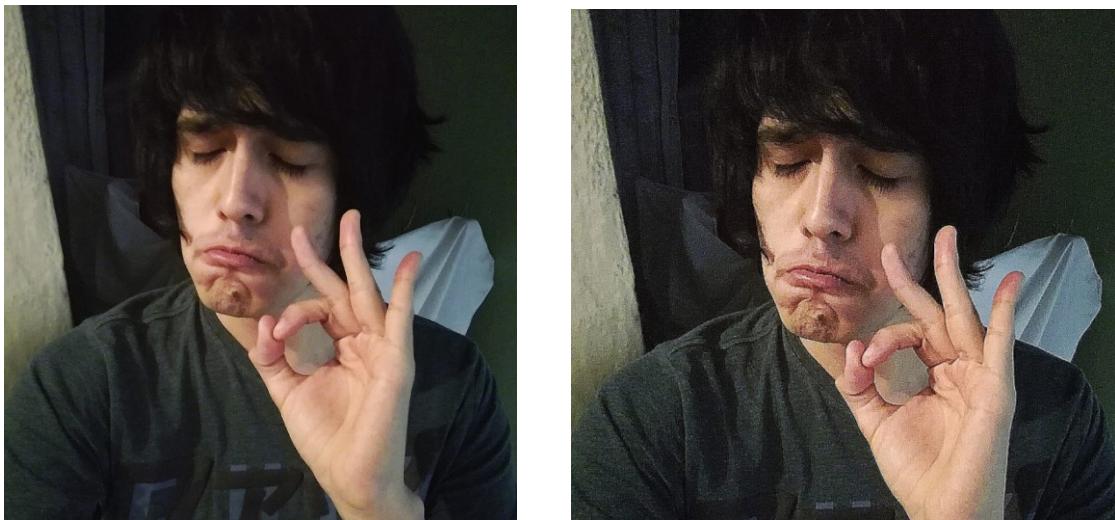


Imagen 20: Foto original con su equivalente tras aplicar el Laplaciano tipo D

### Filtro diferencia de pixeles

El filtro de diferencia de pixeles es un operador usado para la detección de bordes. Utiliza dos kernel diferentes donde uno es usado para el eje horizontal mientras que otro es para el eje vertical.

Está representado por los siguientes Kernel:

$$G_x = \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 1 & -1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad G_y = \begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

Figura 6: Kernel de convolución para filas y columnas del filtro diferencia de pixeles.



Imagen 21: Fotografía tras aplicarse el filtro diferencia de pixeles

### Filtro diferencia de pixeles separados

El filtro de diferencia de pixeles separados es un operador usado para la detección de bordes. Utiliza dos kernel diferentes donde uno es usado para el eje horizontal mientras que otro es para el eje vertical.

Está representado por los siguientes Kernel:

$$Gx = \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 1 & 0 & -1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad Gy = \begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

Figura 7: Kernel de convolución para filas y columnas del filtro diferencia de píxeles separados.



Imagen 22: Fotografía tras aplicarse el filtro diferencia de píxeles separados

### Filtros pasa bajas

La operación de filtrado más sencilla es el filtro pasa bajas. Un filtro de pasa bajas, también llamado filtro de "desenfoque" o "suavizado", promedia cambios rápidos en la intensidad. El filtro pasa bajas más simple simplemente calcula el promedio de un píxel y todos sus ocho vecinos inmediatos. El resultado reemplaza el valor original del píxel repitiendo el proceso para cada píxel en la imagen. [7]



Imagen 23: Imagen tras ser suavizada con un filtro pasa bajas (promedio).

### Filtro promedio

Es el filtro más simple y se basa en el promedio aritmético de los vecinos de un píxel, este filtro tiene las propiedades de suavizar, eliminar ruido y difuminar bordes, además de que se

encarga de atenuar la información de alta frecuencia, dejando intacta la información de baja frecuencia. Tiene un coeficiente de balance  $k = 1/9$

Está representado por el siguiente Kernel:

$$P = \begin{array}{|c|c|c|} \hline +1 & +1 & +1 \\ \hline +1 & +1 & +1 \\ \hline +1 & +1 & +1 \\ \hline \end{array}$$

Figura 8: Kernel de convolución del filtro de Promedio.



Imagen 24: Imagen tras ser suavizada con el filtro promedio.

### Filtro promedio pesado

Este filtro está basado en el filtro promedio, tiene las propiedades de suavizar, eliminar ruido y difuminar bordes, haciéndolo de una manera más suave o débil que el filtro promedio al poner mayor peso al pixel central, además de que se encarga de atenuar la información de alta frecuencia, dejando intacta la información de baja frecuencia. Tiene un coeficiente de balance  $k = 1 / (N + 8)$  donde  $N$  es un valor cambiante elegible por el usuario

Está representado por el siguiente Kernel:

$$P = \begin{array}{|c|c|c|} \hline +1 & +1 & +1 \\ \hline +1 & N & +1 \\ \hline +1 & +1 & +1 \\ \hline \end{array}$$

Figura 9: Kernel de convolución del filtro promedio pesado.



Imagen 25: Imagen tras ser suavizada con el filtro promedio pesado y un N = 10.

### Filtro gaussiano

El filtro Gaussiano se utiliza para desenfocar imágenes y eliminar ruidos y detalles, funciona usando la distribución 2D como punto de propagación. Esto se logra mediante la convolución de la función de distribución gaussiana 2D con la imagen.

La función gaussiana se utiliza, en numerosas áreas de investigación:

- Define una distribución de probabilidad para ruido o datos.
- Es un operador de alisado.
- Se utiliza en matemáticas.
- Está representado por el siguiente Kernel:

$$P = \begin{array}{|c|c|c|} \hline +1 & +1 & +1 \\ \hline +1 & N & +1 \\ \hline +1 & +1 & +1 \\ \hline \end{array}$$

Figura 10: Kernel de convolución del filtro gaussiano.



Imagen 26: Imagen tras ser tratada con el filtro promedio gaussiano

## Filtros no lineales

Los filtros no lineales tienen un comportamiento muy diferente en comparación con los filtros lineales. Para los filtros no lineales, la salida del filtro o la respuesta del filtro no obedecen a los principios que definen un sistema lineal. Los dos primeros son las definiciones básicas de linealidad.

Si se define que un sistema tiene una entrada como  $x[n] = ax[n_1] + bx[n_2]$ , entonces la respuesta del sistema lineal es  $y[n] = ay[n_1] + by[n_2]$ . Esto se conoce como la propiedad de superposición y es fundamental para el diseño del sistema lineal.

La segunda propiedad es la invariancia de turno, si  $y[n]$  es la respuesta a un sistema lineal invariable con cambios con entrada  $x[n]$ , entonces  $y[n-n_0]$  es la respuesta al sistema con entrada  $x[n-n_0]$ .

Además, un filtro no lineal puede producir resultados que varían de una manera no intuitiva. El filtro no lineal más simple a considerar es el filtro de orden de rango o mediana. En el filtro de mediana, la salida del filtro depende del orden de los valores de entrada, generalmente clasificados de menor a mayor o viceversa, se utiliza un rango de soporte de filtro con un número impar de valores, lo que facilita la selección de la salida. [8]

### Filtro media

Es el filtro no lineal más simple y se basa en el promedio aritmético de los vecinos de un píxel, este filtro tiene las propiedades de suavizar, eliminar ruido y difuminar bordes.

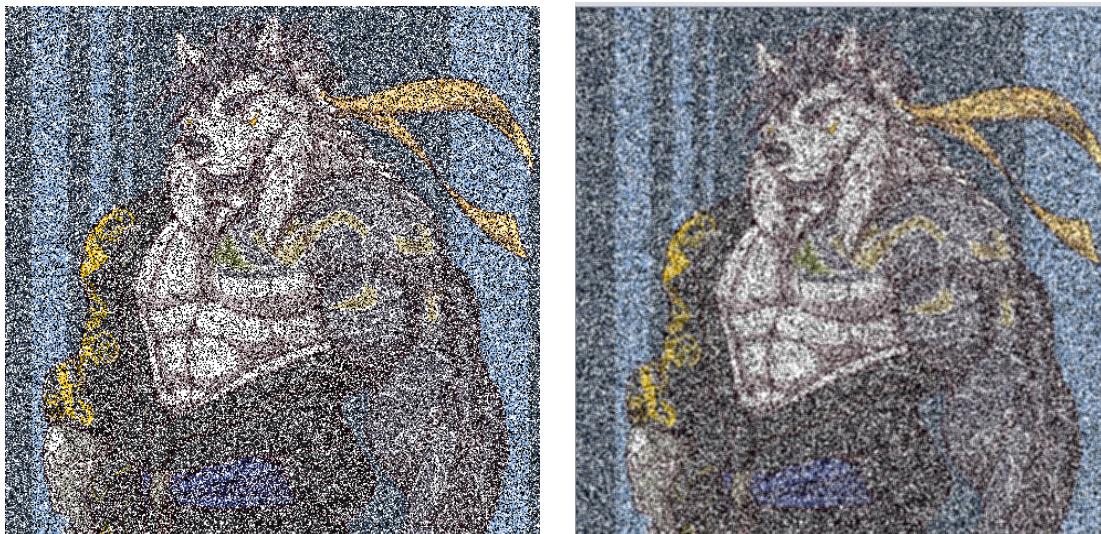


Imagen 27: Imagen tras ser tratada con el filtro media.

### Filtro mediana

El filtro de la mediana es un método no lineal utilizado para eliminar el ruido de imágenes, es ampliamente utilizado ya que es muy efectivo para eliminar el ruido conservando los bordes. Es particularmente efectivo para eliminar el ruido de tipo sal y pimienta.

El filtro mediano funciona moviéndose a través de la imagen píxel por píxel, reemplazando cada valor con el valor de la mediana de píxeles vecinos.

La mediana se calcula ordenando primero todos los valores de píxeles de la ventana o vecindad en orden numérico, y luego reemplazar el valor del píxel por aquel que se encuentra justo en el medio de los valores ordenados, en caso de haber 2 en el centro, se suman y se dividen entre 2.



Imagen 28: Fotografía con ruido sal y pimienta tras ser tratada con el filtro mediana.

### Filtro moda

El filtro de moda se usa para eliminar el ruido de una imagen al reemplazar los píxeles con el valor de píxel que ocurre con mayor frecuencia seleccionado de un determinado tamaño de ventana o vecindad.

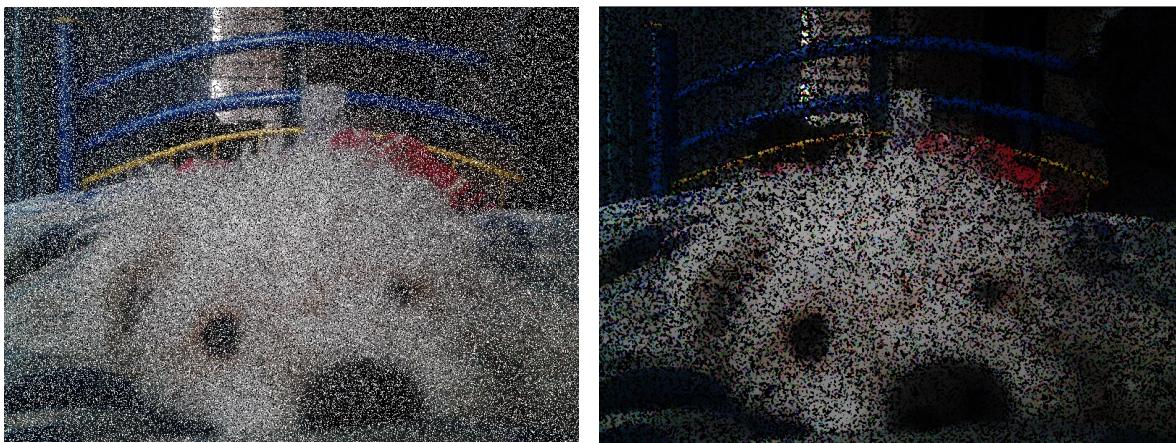


Imagen 29: Fotografía con ruido sal y pimienta tras ser tratada con el filtro moda.

### Filtro máximo

El filtro máximo reemplaza el valor de brillo del pixel central con el valor de brillo más alto existente entre sus pixeles vecinos, ayuda a reducir el ruido tipo pimienta, en caso de haber ruido tipo sal, éste se incrementará.

$$F(x,y) = \begin{array}{|c|c|c|} \hline 20 & 8 & 4 \\ \hline 5 & 10 & 1 \\ \hline 18 & 15 & 6 \\ \hline \end{array}$$

$$G(x,y) = \begin{array}{|c|c|c|} \hline 20 & 8 & 4 \\ \hline 5 & \textcolor{red}{20} & 1 \\ \hline 18 & 15 & 6 \\ \hline \end{array}$$

Figura 11: Ejemplo de transformación tras aplicar el filtro máximo

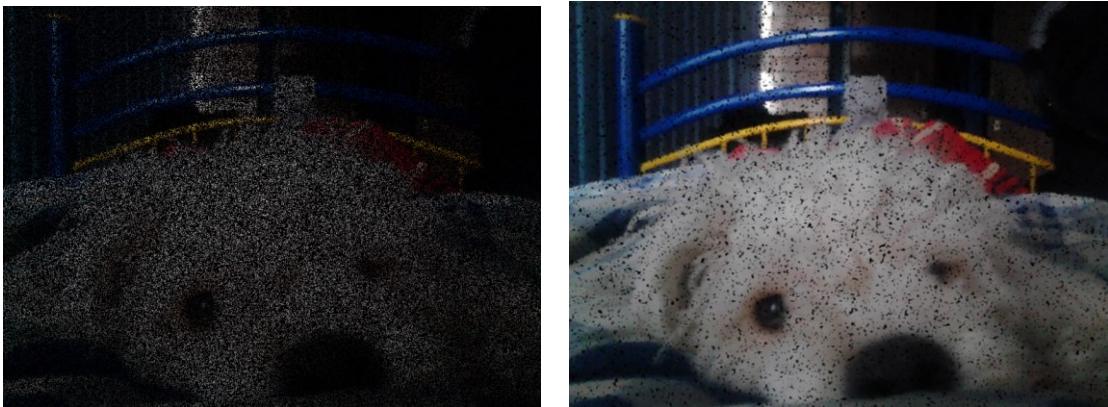


Imagen 30: Fotografía con ruido sal y pimienta tras ser tratada con el filtro máximo.

### Filtro mínimo

El filtro mínimo reemplaza el valor de brillo del pixel central con el valor de brillo más bajo existente entre sus pixeles vecinos, ayuda a reducir el ruido tipo sal, en caso de haber ruido tipo pimienta, éste se incrementará.

$$F(x,y) = \begin{array}{|c|c|c|} \hline 20 & 8 & 4 \\ \hline 5 & 10 & 1 \\ \hline 18 & 15 & 6 \\ \hline \end{array}$$

$$G(x,y) = \begin{array}{|c|c|c|} \hline 20 & 8 & 4 \\ \hline 5 & \textcolor{red}{1} & 1 \\ \hline 18 & 15 & 6 \\ \hline \end{array}$$

Figura 12: Ejemplo de transformación tras aplicar el filtro mínimo.



Imagen 31: Fotografía con ruido sal y pimienta tras ser tratada con el filtro mínimo.

## Segmentación global

La segmentación de imágenes se encarga de descomponer una imagen en sus partes constituyentes, es decir, los objetos de interés y el fondo, basándose en ciertas características locales que permiten distinguir un objeto del fondo y objetos entre sí. La mayoría de las imágenes están constituidas por regiones o zonas que tienen características homogéneas, por ejemplo, el nivel de gris, la textura, etc., generalmente estas regiones corresponden a objetos de la imagen.

La segmentación de una imagen consiste en la división o partición de la imagen en varias zonas o regiones homogéneas y disjuntas a partir de su contorno, su conectividad, o en términos de un conjunto de características de los píxeles de la imagen que permitan discriminar unas regiones de otras.

Los tonos de gris, la textura, la magnitud del gradiente, la dirección de los bordes, las modas de los tonos de gris en ventanas 3x3, 7x7 y 15x15, etc., son algunas características a utilizar para la segmentación.

La segmentación completa aparece cuando las regiones disjuntas corresponden directamente a objetos de la imagen mientras que la segmentación parcial, cuando las regiones no se corresponden directamente con objetos de la imagen. Para conseguir la segmentación completa se necesita un nivel superior de conocimiento que utiliza un conocimiento específico del dominio de la escena, este conocimiento puede ser, por ejemplo, que los objetos de la imagen corresponden a caracteres numéricos o letras de un alfabeto. La operación de segmentación trata de distinguir si un píxel pertenece, o no, a un objeto de interés y, por lo tanto, produce una imagen binaria. Algunos algoritmos de segmentación de imágenes se basan en alguna de las tres propiedades siguientes:

- a) Discontinuidad en los tonos de gris de los píxeles de un entorno  
Permite detectar puntos aislados, líneas y aristas (bordes).
- b) Similaridad en los tonos de gris de los píxeles de un entorno  
Permite construir regiones por división y fusión, por crecimiento o por umbralización.
- c) Conectividad de los píxeles  
Una región D se dice conexa o conectada si para cada par de píxeles de la región existe un camino formado por píxeles de D que los conecta. Un camino de píxeles es una secuencia de píxeles adyacentes (que pertenecen a su entorno inmediato).

Los métodos de segmentación se pueden agrupar en cuatro clases diferentes:

- a) Métodos basados en píxeles  
A su vez pueden ser:
  - Locales  
Basadas en las propiedades de los píxeles y su entorno

- Globales  
Basadas en la información global obtenida, por ejemplo, con el histograma de la imagen.
- b) Métodos basados en bordes.  
c) Métodos basados en regiones, que utilizan las nociones de homogeneidad y proximidad geométrica, como las técnicas de crecimiento, fusión o división.  
d) Métodos basados en modelos. [9]

## Binarización

La binarización de una imagen consiste en un proceso de reducción de la información de la misma, en la que sólo persisten dos valores: verdadero y falso. En una imagen digital, estos valores, verdadero y falso, pueden representarse por los valores 0 y 1 o, más frecuentemente, por los colores negro (valor de gris 0) y blanco (valor de gris 255).

La binarización se emplea para separar las regiones u objetos de interés en una imagen del resto. Las imágenes binarias se usan en operaciones booleanas o lógicas para identificar individualmente objetos de interés o para crear máscaras sobre regiones. En muchos casos, una imagen binaria es el resultado de una segmentación por niveles de gris o de una segmentación por selección de un rango de color determinado. [10]



Imagen 32: Ejemplo de imagen binaria con umbral  $u = 90$

## Binarización de Otsu

La binarización por el método de Otsu es un procedimiento que selecciona el umbral óptimo maximizando la varianza entre clases mediante una búsqueda exhaustiva. [12]

Para realizar la binarización se parte de una imagen en niveles de gris con N píxeles y L posibles niveles de gris diferentes, expresando mediante la fórmula:

$$p_i = \frac{f_i}{N}$$

Donde:

$P_i$  = Probabilidad de ocurrencia del nivel de gris i en la imagen.

$f_i$  = Frecuencia de repetición del nivel de gris i-ésimo con  $i = 1, 2, \dots, L$ .

N = Total de pixeles en la imagen

Posteriormente, los píxeles se dividen en dos clases  $C_1$  y  $C_2$ , con niveles de gris  $[1, 2, \dots, t]$  y  $[t+1, t+2, \dots, L]$  respectivamente, donde las distribuciones de probabilidad de ambas clases son:

$$\begin{aligned} C_1 : & \quad \frac{p_1}{\omega_1(t)}, \dots, \frac{p_t}{\omega_1(t)} \\ C_2 : & \quad \frac{p_{t+1}}{\omega_2(t)}, \frac{p_{t+2}}{\omega_2(t)}, \dots, \frac{p_L}{\omega_2(t)} \end{aligned}$$

Y donde:

$$\omega_1(t) = \sum_{i=1}^t p_i \qquad \omega_2(t) = \sum_{i=t+1}^L p_i$$

El cual es conocido como momento acumulado de orden cero de la k-ésima clase  $C_k$ .

Posteriormente se calcula la media para cada clase mediante la expresión:

$$\mu_1 = \sum_{i=1}^t \frac{i \cdot p_i}{\omega_1(t)} \qquad \mu_2 = \sum_{i=t+1}^L \frac{i \cdot p_i}{\omega_2(t)}$$

Haciendo uso de un análisis discriminante, Otsu definió la varianza entre clases de una imagen umbralizada como:

$$\sigma_B^2 = \omega_1 \cdot (\mu_1 - \mu_T)^2 + \omega_2 \cdot (\mu_2 - \mu_T)^2$$

Donde la intensidad media total es:

$$\omega_1 \cdot \mu_1 + \omega_2 \cdot \mu_2 = \mu_T$$

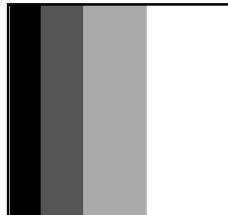
Otsu demostró que el umbral óptimo es:

$$t^* = \underset{t}{\operatorname{Max}} \{\sigma_B^2(t)\}$$

Donde:

$$1 < t < L$$

Consideremos la siguiente imagen con los siguientes parámetros que la definen.



Niveles de Gris:  $L = 4$  [0, 85, 171, 255]

Frecuencias de aparición:  $f_1 = 10, f_2 = 20, f_3 = 30, f_4 = 40$

Total de pixeles:  $N = 100$  (10x10)

Dividimos los niveles de gris en dos clases:

$$C1 = [0, 85] \text{ y } C2 = [171, 255]$$

Calculamos la probabilidad de ocurrencia para cada nivel de gris de la imagen

$$p_1 = \frac{f_1}{N} = \frac{10}{100} = \frac{1}{10}$$

$$p_2 = \frac{f_2}{N} = \frac{20}{100} = \frac{1}{5}$$

$$p_3 = \frac{f_3}{N} = \frac{30}{100} = \frac{3}{10}$$

$$p_4 = \frac{f_4}{N} = \frac{40}{100} = \frac{4}{10}$$

$$\omega_1(t=2) = \sum_{i=1}^t p_i = \frac{1}{10} + \frac{1}{5} = \frac{3}{10}$$

$$\omega_2(t=2) = \sum_{i=t+1}^L p_i = \frac{3}{10} + \frac{4}{10} = \frac{7}{10}$$

Se calcula la distribución de probabilidad para cada una de las clases

$$C_1: \frac{p_1}{\omega_1(t=2)}, \frac{p_2}{\omega_1(t=2)} = \left\{ \frac{1}{3}, \frac{2}{3} \right\}$$

$$C_2: \frac{p_3}{\omega_2(t=2)}, \frac{p_4}{\omega_2(t=2)} = \left\{ \frac{3}{7}, \frac{4}{7} \right\}$$

Posteriormente calculamos la media para ambas clases

$$\mu_1 = \sum_{i=1}^t \frac{ip_i}{\omega_1(t=2)} = \frac{1}{3} + 2 \times \frac{2}{3} = \frac{5}{3}$$

$$\mu_2 = \sum_{i=t+1}^L \frac{ip_i}{\omega_2(t=2)} = 3 \times \frac{3}{7} + 4 \times \frac{4}{7} = \frac{25}{7}$$

Obtenemos la intensidad media total

$$\mu_T = \omega_1(t=2)\mu_1 + \omega_2(t=2)\mu_2 = 3$$

Calculamos la varianza de las clases para conocer aquel umbral que nos otorgue el mayor valor de varianza

$$\sigma_B^2 = \omega_1(t=2)(\mu_1 - \mu_T)^2 + \omega_2(t=2)(\mu_2 - \mu_T)^2 \approx 0.7619$$

$$\sigma_B^2(t=1) = 0.4444$$

$$\sigma_B^2(t=3) = 0.6667$$

Tras calcular para todos los niveles de gris obtenemos que el umbral ideal es aquel umbral  $t = 2$  que corresponde al nivel de gris 85

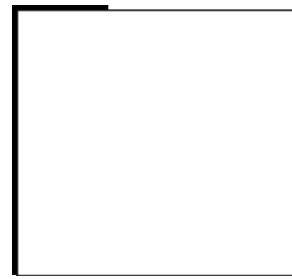
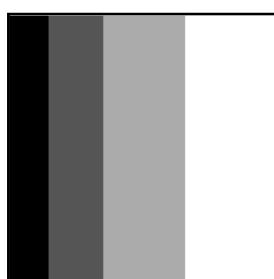


Imagen 32.2 Imagen de los cuatro niveles de gris binarizada con Otsu.

## Umbralización

En muchas aplicaciones, es útil poder separar las regiones de la imagen correspondientes a los objetos en los que estamos interesados, de las regiones de la imagen que corresponden al fondo. El umbral o grupo de umbrales a menudo proporcionan una forma fácil y conveniente de realizar esta segmentación en función de las diferentes intensidades o colores en las regiones de primer plano y fondo de una imagen.

Además, es útil poder ver qué áreas de una imagen constan de píxeles cuyos valores se encuentran dentro de un rango específico, o banda de intensidades (o colores).

En su funcionamiento, la entrada suele ser una imagen en escala de grises o en color, en la implementación más simple, la salida es una imagen binaria que representa la segmentación. Los píxeles negros corresponden al fondo y los píxeles blancos corresponden al primer plano u objetos (o viceversa). La segmentación está determinada por un único parámetro conocido como el umbral de intensidad, en donde cada píxel de la imagen se compara con este umbral, si la intensidad del píxel es mayor que el umbral, el píxel se establece blanco en la salida, si es menor que el umbral, se establece en negro.

En implementaciones más sofisticadas, se pueden especificar múltiples umbrales, de modo que una banda de valores de intensidad se puede establecer en blanco, mientras que todo lo demás se establece en negro. Para imágenes en color o multiespectrales, puede ser posible establecer umbrales diferentes para cada canal de color, y así seleccionar solo esos píxeles dentro de un cuboide especificado en el espacio RGB. Otra variante común es configurar en negro todos los píxeles correspondientes al fondo, pero dejar los píxeles del primer plano en su color / intensidad original (en lugar de forzarlos a blanco), para que esa información no se pierda. [11]



Imagen 33: Imagen con 5 umbrales definidos en  $U_1 = 10$ ,  $U_2 = 20$ ,  $U_3 = 30$ ,  $U_4 = 40$ ,  $U_5 = 50$

## Segmentación de una imagen

Extracción de las bolas de billar, el palo, el encerador y la tetera.

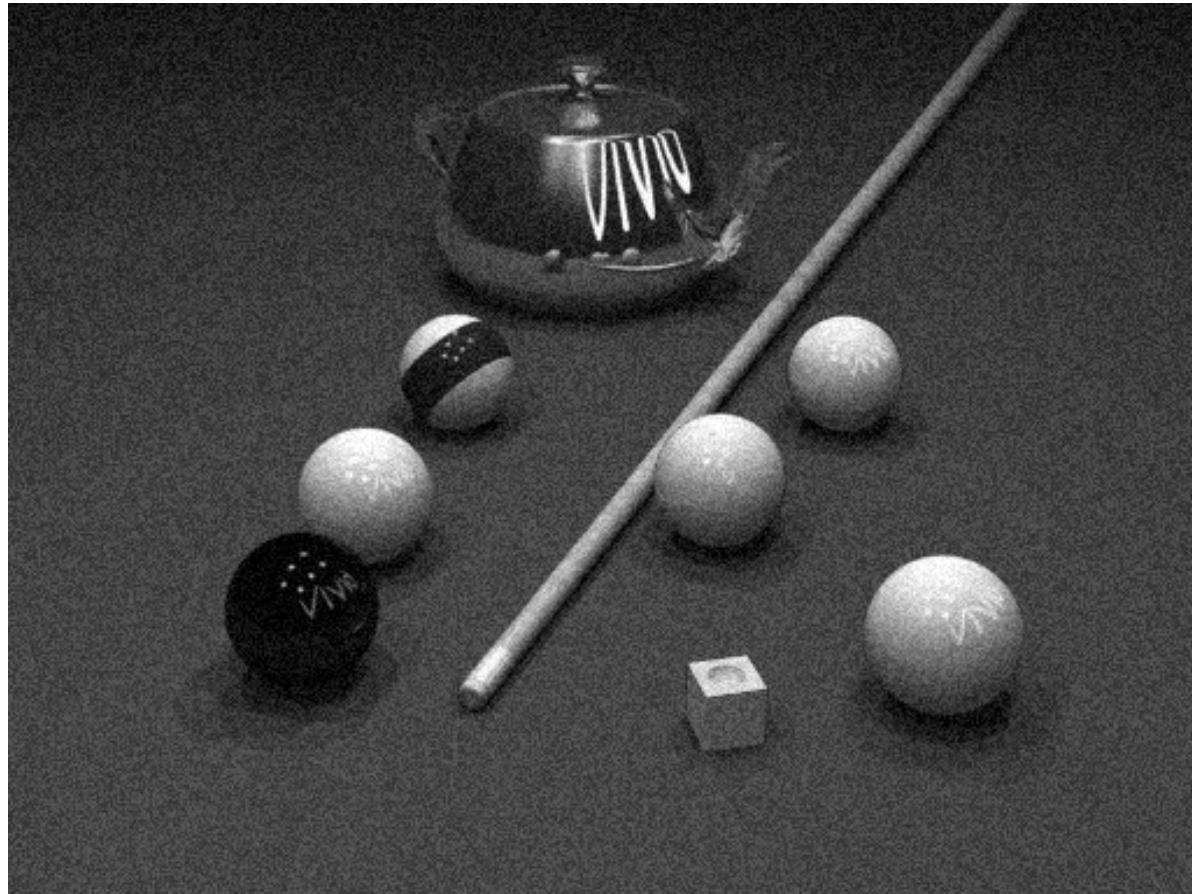


Imagen 34: Imagen original de una mesa de billar y las bolas.

Paso 1:

Aplicar al filtro máximo 3 veces para eliminar el ruido tipo pimienta.

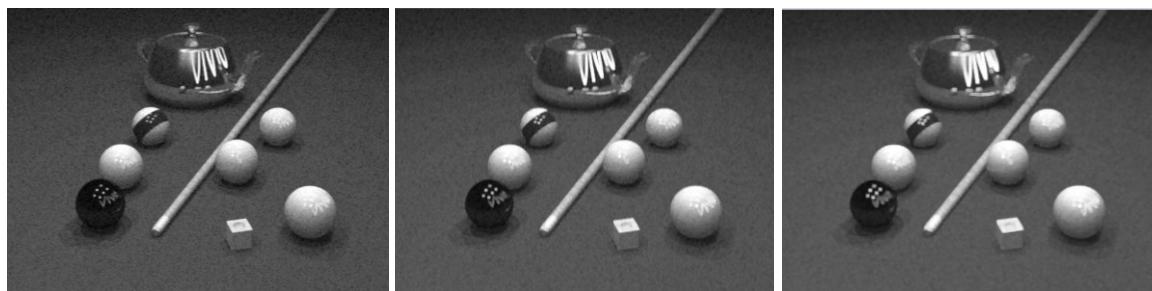


Imagen 35: Aplicación del filtro máximo sobre la imagen original, primera, segunda y tercera aplicación del filtro respectivamente.

## Paso 2: Multiumbralización

Aplicando un umbral entre 0 – 30 tenemos el siguiente resultado



Imagen 36: Resultado tras el umbralizado de la imagen con  $U = 0$  y  $U_2 = 30$

Aplicando un umbral de 100 a 110

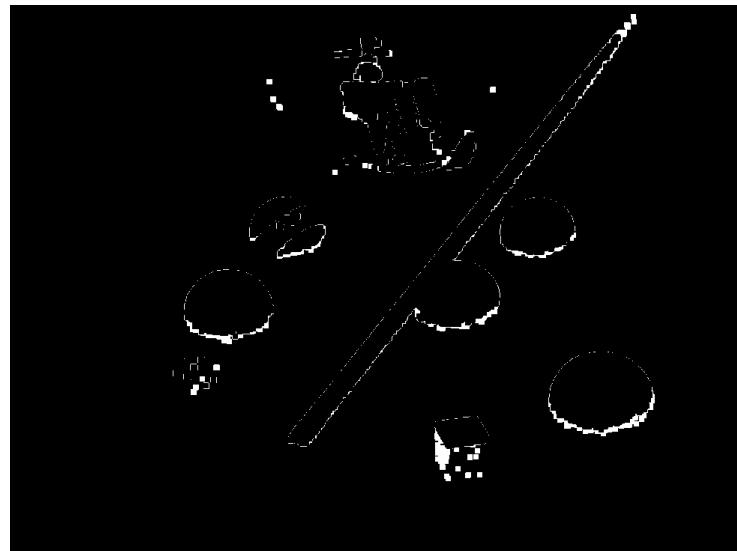


Imagen 37: Resultado tras el umbralizado de la imagen con  $U = 100$  y  $U_2 = 110$

Aplicando un umbral de 100 a 255

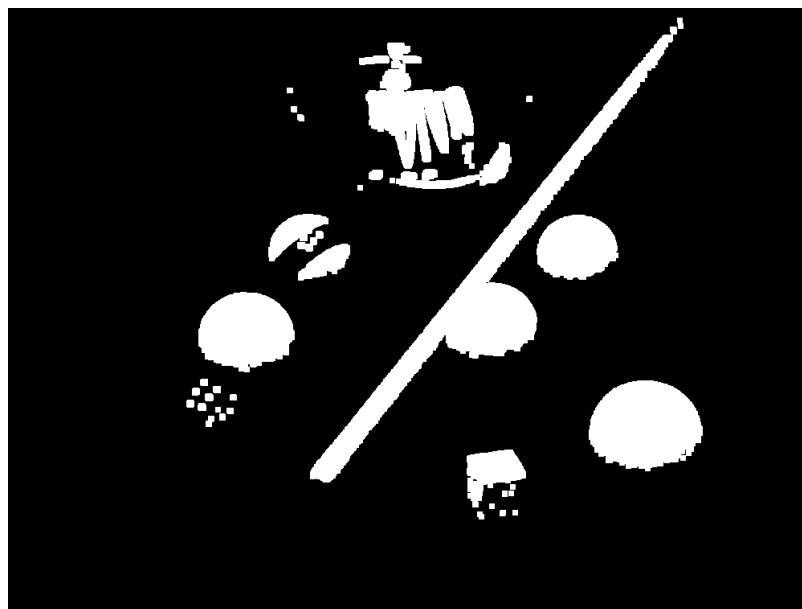


Imagen 38: Resultado tras el umbralizado de la imagen con  $U = 100$  y  $U_2 = 255$

Sumando las imágenes que contienen los umbrales 0 – 30 y 100 – 255

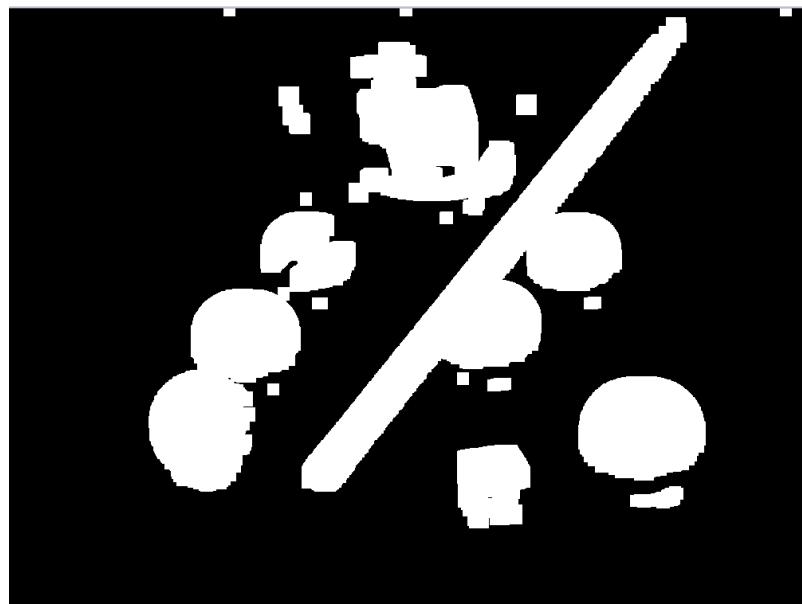


Imagen 39: Resultado tras la suma de las imágenes 36, 37 y 38

### Pase 3: Filtrado

A la imagen limpia le aplicamos el filtro de diferencia de píxeles separados

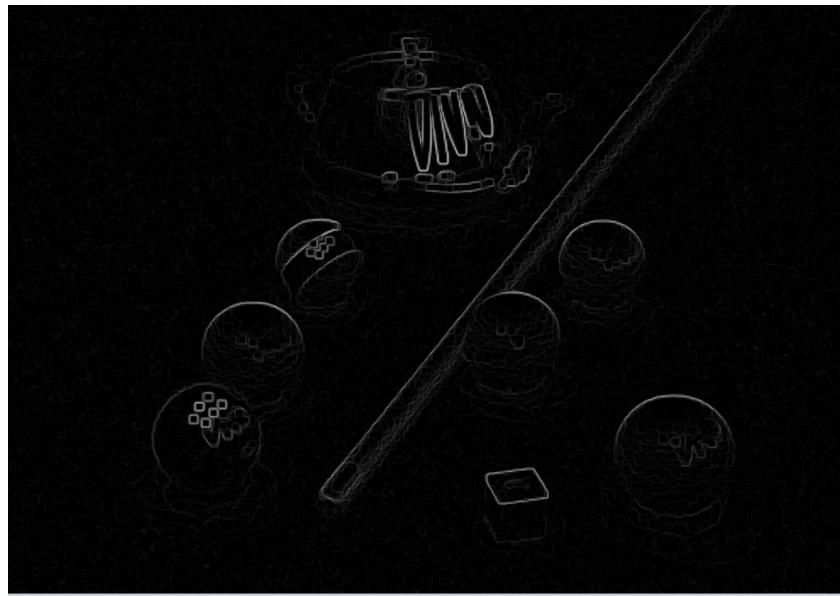


Imagen 40: Imagen limpia tras aplicarle el filtro diferencia de píxeles separados

Le aplicamos el filtro mínimo

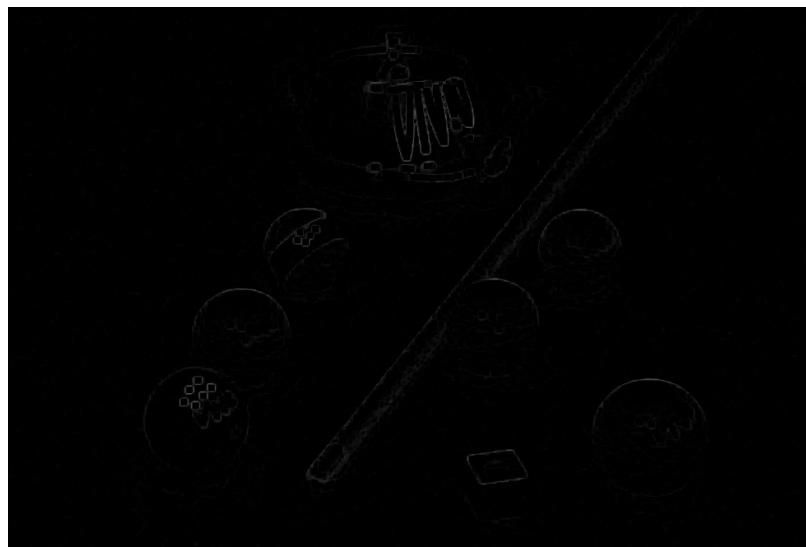


Imagen 41: Imagen 40 tras aplicar el filtro mínimo para eliminar ruido sal.

Binarizamos con un umbral  $u = 10$

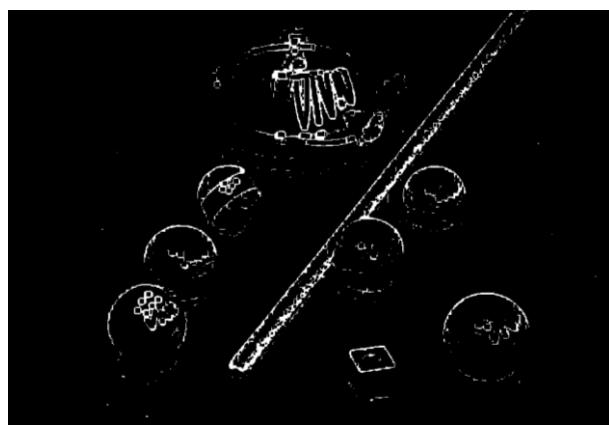


Imagen 42: Aplicación de un umbral  $U = 10$

Paso 4: Unión

Sumamos con la imagen para tener una mejor definición de los objetos

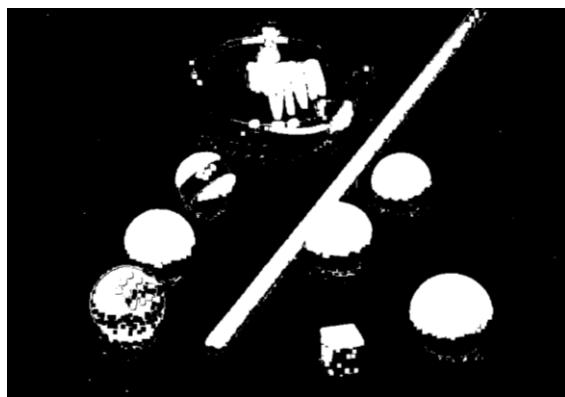


Imagen 43: Suma de la imagen 42 y la imagen 39

Hacemos una resta, donde a la imagen original se le resta la imagen para ver un  
aproximado de la segmentación

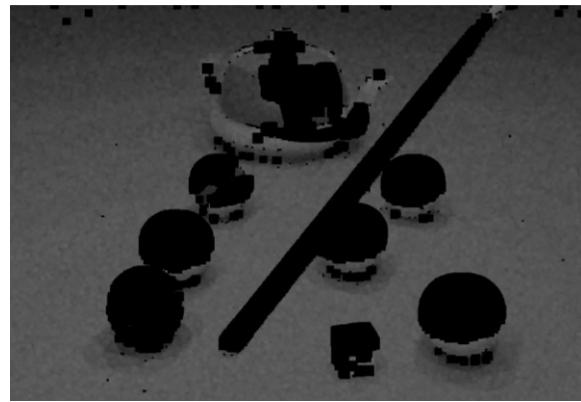


Imagen 44: Segmentación aproximada de los objetos restando la original con la imagen 43.

## Conclusiones

Tras la aplicación de los diferentes filtros, operaciones sobre el histograma y métodos de segmentación nos damos cuenta de que las operaciones nos ayudan a mejorar el contraste de una imagen operando sobre el histograma, aumentando o reduciendo el brillo a consideración del usuario o incluso repartiéndolo entre todos los niveles de gris del histograma, permitiendo tener una imagen más clara y con colores o niveles de gris más definidos.

Por otra parte, los filtros nos permiten desde suavizar una imagen para poder eliminar el ruido, hasta la obtención de los bordes de los objetos de la misma, dentro de los filtros suavizantes tenemos el filtro promedio, promedio pesado, gaussiano, mediana, moda donde el filtro promedio pesado nos permite un suavizado más leve a comparación de su homónimo ya que el valor del pixel central se incrementa, el filtro gaussiano nos da un difuminado muy ligeramente mayor pero conserva ligeramente mejor los bordes ya que trabaja con un kernel el cual tiene más peso en el centro y en sus vecinos de la 4 vecindad mientras que sus esquinas no son aumentadas de valor. En los filtros para la obtención de bordes, el filtro de Sobel nos da un resultado ligeramente mayor en grosor ya que su kernel tiene 2 vecinos horizontales y 2 verticales en los cuales el valor se duplica mientras que el filtro de Prewitt, el filtro de Roberts nos da bordes más claros, pero niveles de gris más intensos ya que utiliza solo un vecino diagonal de los pixeles.

También tenemos los filtros no lineales como son la media, moda, máximo y mínimo de los cuales no podemos predecir o conocer su resultado hasta después de aplicarlos ya que éstos filtros dependen de los valores de gris de la imagen y varían de imagen a imagen debido a que son filtros probabilísticos que buscan reducir el ruido, pero el coste de su uso es la ligera difuminación de la imagen, los filtros máximo y mínimo retornan el máximo o mínimo local

Por último, tenemos la binarización, la cual coloca el valor de 255 a aquellos valores de gris que estén entre un grupo de umbrales en el caso de la multiumbralización o sean mayor al umbral para la binarización simple, colocando a su vez el valor 0 para aquellos que no cumplen dicha condición, permitiéndonos así separar e identificar de manera clara y sencilla un objeto o grupo de objetos separados o segmentados únicamente por su nivel de brillo.

## Referencias

- [1]. A.D Jepson and D.J Fleet, "Computer Science," *Department of Computer Science, University of Toronto*, 2007. [Online]. Available: <http://www.cs.toronto.edu/>. [Accessed: 31-Oct-2018].
- [2]. Nick Efford. *Digital Image Processing: A Practical Introduction Using Java<sup>TM</sup>*. Pearson Education, 2000.
- [3]. "conv2," What Is Image Filtering in the Spatial Domain? - MATLAB & Simulink - MathWorks América Latina. [Online]. Available: <https://la.mathworks.com/help/images/what-is-image-filtering-in-the-spatial-domain.html>. [Accessed: 02-Nov-2018].
- [4]. Aziz Makandar and Bhagirathi Halalli, "Image Enhancement Techniques using Highpass and Lowpass Filters," Image Enhancement Techniques using Highpass and Lowpass Filters, 14-Jan-2015. [Online]. Available: <https://pdfs.semanticscholar.org/9b10/febd207e92f9d362ed06286d0ba31e290fec.pdf>. [Accessed: 01-Nov-2018].
- [5]. "High-Pass Filtering (Sharpening)," High-Pass Filtering (Sharpening). [Online]. Available: [https://diffractionlimited.com/help/maximdl/High-Pass\\_Filtering.htm](https://diffractionlimited.com/help/maximdl/High-Pass_Filtering.htm). [Accessed: 02-Nov-2018].
- [6]. Fr. Don Matthys, "Laplacian of Gaussian Filter," Laplacian of Gaussian Filter, 21-Mar-2001. [Online]. Available: <https://academic.mu.edu/phys/matthysd/web226/Lab02.htm>. [Accessed: 02-Nov-2018].
- [7]. Filtering an Image, 16-Jun-2005. [Online]. Available: [http://northstar-www.dartmouth.edu/doc/idl/html\\_6.2/Filtering\\_an\\_Imagehvr.html](http://northstar-www.dartmouth.edu/doc/idl/html_6.2/Filtering_an_Imagehvr.html). [Accessed: 02-Nov-2018].
- [8]. "Comparing linear versus nonlinear filters in image processing," Embedded Computing Design. [Online]. Available: <http://www.embedded-computing.com/embedded-computing-design/comparing-linear-versus-nonlinear-filters-in-image-processing>. [Accessed: 03-Nov-2018].

- [9]. "LCC:: Departamento de Lenguajes y Ciencias de la Computación," LCC :: Departamento de Lenguajes y Ciencias de la Computación. [Online]. Available: <http://www.lcc.uma.es/>. [Accessed: 08-Nov-2018].
- [10]. L. Sol, "Binarizacion de las imágenes," Imagen binaria. [Online]. Available: [http://www.dimages.es/Tutorial\\_A.I/segmentacion/binaria.htm](http://www.dimages.es/Tutorial_A.I/segmentacion/binaria.htm). [Accessed: 08-Nov-2018].
- [11]. "Thresholding," Point Operations - Thresholding. [Online]. Available: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/threshld.htm>. [Accessed: 08-Nov-2018].
- [12]. J. A. P. Serquen, "Segmentación por umbralización método de otsu," LinkedIn SlideShare, 21-Sep-2016. [Online]. Available: <https://www.slideshare.net/JorgeAntonioParraSerquen/segmentacin-por-umbralizacin-mtodo-de-otsu>. [Accessed: 28-Nov-2018].