



CENTRO DE ENSEÑANZA TECNICA INDUSTRIAL PRACTICAS DE LABORATORIO

INGENIERÍA CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
INGENIERÍA DE DESARROLLO DE SOFTWARE	2025-B	19SDS32	PROCESAMIENTO DE IMÁGENES
ALUMNO		FECHA	EVALUACIÓN

PRÁCTICA No.	LABORATORIO DE COMPUTACIÓN No	NOMBRE DE LA PRÁCTICA	DURACIÓN (HORAS)
4	LCS	FILTROS ESPECIALES	2

INTRODUCCIÓN

El procesamiento de imágenes digitales es una disciplina que abarca técnicas y algoritmos para la manipulación y análisis de imágenes. Uno de los aspectos más importantes en esta área es el uso de filtros para modificar la imagen de manera controlada, ya sea para mejorar su calidad, resaltar características específicas o extraer información relevante. Los filtros se aplican mediante convoluciones que permiten transformar la imagen original en una versión filtrada según el propósito deseado.

Existen diversos tipos de filtros que cumplen diferentes funciones, cada uno de estos filtros tiene propiedades y características únicas que los hacen adecuados para distintos tipos de aplicaciones en el procesamiento de imágenes, como la reducción de ruido, el resaltado de bordes y la mejora del contraste.

El filtro de suavizado se emplea comúnmente para eliminar el ruido y suavizar las transiciones de color en la imagen, proporcionando un efecto visual más uniforme y menos estridente. El filtro box es una versión simplificada del filtro de suavizado que utiliza un promedio uniforme de los píxeles circundantes. El filtro gaussiano, por su parte, realiza un suavizado controlado mediante una distribución gaussiana, lo que permite preservar mejor las estructuras importantes en la imagen. Por último, el filtro laplaciano se utiliza principalmente para la detección de bordes, realzando las áreas donde la intensidad cambia rápidamente, y es fundamental en técnicas de segmentación y reconocimiento de patrones.

OBJETIVO (COMPETENCIA)

El objetivo de esta práctica es implementar y aplicar diferentes tipos de filtros de imagen en MATLAB, como el suavizado, el box, el Gauss y el Laplace, para observar sus efectos en una imagen de entrada. Se busca comprender cómo cada filtro afecta la calidad, el ruido y los detalles de la imagen, y cómo seleccionar el filtro adecuado dependiendo del propósito deseado, ya sea reducción de ruido, detección de bordes u otros.

FUNDAMENTO

Filtro de Suavizado

El filtro de suavizado, también conocido como filtro de media, es una técnica que reduce las fluctuaciones bruscas en la intensidad de la imagen. Consiste en reemplazar cada píxel con el promedio de los píxeles vecinos dentro de una ventana definida. Este proceso reduce el ruido de alta frecuencia, como puntos o imperfecciones en la imagen, al promediar las intensidades de los píxeles vecinos. Es ampliamente utilizado en aplicaciones donde se busca una representación visual más homogénea, eliminando detalles menores que pueden considerarse ruido.

Filtro Box

El filtro box es una forma específica de filtro de suavizado en el que se utiliza una ventana rectangular donde todos los píxeles tienen el mismo peso. Es un caso particular del filtro de media y se aplica sumando los valores de los píxeles dentro de la ventana y dividiendo por el número total de píxeles. Este tipo de filtro es muy eficiente computacionalmente y es útil para aplicaciones donde se requiere un suavizado rápido, aunque puede producir efectos de borrosidad en los bordes debido a su enfoque uniforme.

Filtro Gaussiano

El filtro gaussiano se basa en la función de distribución gaussiana, una función continua y simétrica que asigna pesos mayores a los píxeles cercanos al centro de la ventana y pesos menores a los píxeles más alejados. Este filtro es ideal para la reducción de ruido, ya que proporciona un suavizado más suave y natural en comparación con el filtro box. La función gaussiana garantiza que el suavizado sea isotrópico, es decir, que se aplique de manera uniforme en todas las direcciones, preservando mejor las características significativas de la imagen como bordes y texturas.

Filtro Laplaciano

El filtro laplaciano es un filtro de detección de bordes que se basa en el operador laplaciano, que calcula la segunda derivada de la intensidad de la imagen. A diferencia de los filtros de suavizado, que reducen el ruido, el filtro laplaciano se utiliza para resaltar áreas de cambio rápido de intensidad, como los bordes de los objetos. Esto se logra

encontrando las regiones donde la intensidad varía bruscamente, produciendo una imagen que resalta contornos y bordes. Es un componente clave en la detección de características y la segmentación de imágenes, donde es crucial identificar los límites de las estructuras presentes en la imagen.

Estos filtros, aunque simples en su concepto, forman la base para técnicas más avanzadas de procesamiento de imágenes. Cada uno ofrece un enfoque diferente para la manipulación de las imágenes y su correcta aplicación depende del tipo de información que se desea extraer o resaltar.

METODOLOGÍA (DESARROLLO DE LA PRACTICA)

```

Im = imread('carro.png');
In = rgb2gray(Im);
[FIL, COL] = size(In);
% -----Filtro suavizado-----
% -----Filtro suavizado-----
% -----Filtro suavizado-----
% -----Filtro suavizado-----
OutSuavizado = zeros(FIL, COL);
filtro_suavizado = ones(3, 3) / 9;
for rxp = 2:FIL-1
    for ryp = 2:COL-1
        vecindad = double(In(rxp-1:rxp+1, ryp-1:ryp+1));
        nuevo_valor = sum(sum(vecindad .* filtro_suavizado));
        OutSuavizado(rxp, ryp) = nuevo_valor;
    end
end
OutSuavizado = uint8(OutSuavizado);
% -----Filtro box-----
% -----Filtro box-----
% -----Filtro box-----
% -----Filtro box-----
OutBox = zeros(FIL, COL);
filtro_box = [0,0,0,0,0;0,1,1,1,0;0,1,1,1,0;0,1,1,1,0;0,0,0,0,0]/9;
for rxp = 3:FIL-2
    for ryp = 3:COL-2
        vecindad = double(In(rxp-2:rxp+2, ryp-2:ryp+2));
        nuevo_valor = sum(sum(vecindad .* filtro_box));
        OutBox(rxp, ryp) = nuevo_valor;
    end
end
OutBox(1, :) = 0; % Primera fila
OutBox(FIL, :) = 0; % Última fila

```

```

OutBox(:, 1) = 0; % Primera columna
OutBox(:, COL) = 0; % Última columna
OutBox = uint8(OutBox);
% -----Filtro gaussiano-----
% -----%
% -----Filtro gaussiano-----
% -----%
% -----Filtro gaussiano-----
% -----%
OutGauss = zeros(FIL, COL);
sigma = 1;
[x, y] = meshgrid(-2:2, -2:2); % Mascara de 5x5
filtro_gaussiano = exp(-(x.^2 + y.^2) / (2 * sigma^2)); % Formula gaussiana
filtro_gaussiano = filtro_gaussiano / sum(filtro_gaussiano(:)); % Normalizar la
mascara
for rxp = 3:FIL-2
    for ryp = 3:COL-2
        % Vecindad 5x5 del píxel actual (rxp, ryp)
        vecindad = double(In(rxp-2:rxp+2, ryp-2:ryp+2));

        % Aplicar la máscara gaussiana
        nuevo_valor = sum(sum(vecindad .* filtro_gaussiano));

        % Asignar el valor calculado al píxel correspondiente en la imagen de
salida
        OutGauss(rxp, ryp) = nuevo_valor;
    end
end
OutGauss = uint8(OutGauss);
% -----Filtro Laplaciano-----
% -----%
% -----Filtro Laplaciano-----
% -----%
% -----Filtro Laplaciano-----
% -----%
OutLaplaciano = zeros(FIL, COL);
filtro_laplaciano = [ 0  0 -1  0  0;
                     0 -1 -2 -1  0;
                    -1 -2 16 -2 -1;
                     0 -1 -2 -1  0;
                     0  0 -1  0  0];
for rxp = 3:FIL-2
    for ryp = 3:COL-2
        % Vecindad 5x5 del píxel actual (rxp, ryp)
        vecindad = double(In(rxp-2:rxp+2, ryp-2:ryp+2));

        % Aplicar la máscara Laplaciana
        nuevo_valor = sum(sum(vecindad .* filtro_laplaciano));

        % Asignar el valor calculado al píxel correspondiente en la imagen de
salida
        OutLaplaciano(rxp, ryp) = nuevo_valor;
    end
end

```

```

    end
end
OutLaplaciano = uint8(OutLaplaciano);
% Mostrar la imagen original y los resultados de los filtros en una misma
ventana (figure)
figure;
imshow(In);
title('Imagen Original');
% Mostrar los resultados de los cuatro filtros en otra ventana
figure;
% Mostrar la imagen suavizada (Media 3x3) en la primera fila, primera columna
subplot(2, 2, 1); % Posición 1 de la cuadrícula 2x2
imshow(OutSuavizado);
title('Imagen Suavizada (Media 3x3)');
% Mostrar la imagen con el filtro Box aplicado en la primera fila, segunda
columna
subplot(2, 2, 2); % Posición 2 de la cuadrícula 2x2
imshow(OutBox);
title('Filtro Box con Bordes Negros');
% Mostrar la imagen con el filtro Gaussiano aplicado en la segunda fila, primera
columna
subplot(2, 2, 3); % Posición 3 de la cuadrícula 2x2
imshow(OutGauss);
title('Filtro Gaussiano 5x5 (sigma=1)');
% Mostrar la imagen con el filtro Laplaciano aplicado en la segunda fila,
segunda columna
subplot(2, 2, 4); % Posición 4 de la cuadrícula 2x2
imshow(OutLaplaciano);
title('Filtro Laplaciano 5x5');

```

1. Lectura y Preparación de la Imagen

```

Im = imread('gatito.png');
In = rgb2gray(Im);
[FIL, COL] = size(In);

```

En esta parte del código, se carga la imagen gatito.png en la variable Im usando la función imread. La imagen se convierte a escala de grises utilizando la función rgb2gray, y el resultado se almacena en In. Esto se hace porque los filtros generalmente se aplican a imágenes en escala de grises para simplificar el procesamiento. Luego, se obtienen las dimensiones de la imagen, almacenándose el número de filas en FIL y el número de columnas en COL, lo que permite iterar sobre la imagen más adelante en el código.

2. Filtro de Suavizado (Media 3x3)

```

OutSuavizado = zeros(FIL, COL);
filtro_suavizado = ones(3, 3) / 9;
for rpx = 2:FIL-1

```

```

for ryp = 2:COL-1
    % Vecindad 3x3 del píxel actual (rxp, ryp)
    vecindad = double(In(rxp-1:rxp+1, ryp-1:ryp+1));

    % Aplicar la máscara de suavizado
    nuevo_valor = sum(sum(vecindad .* filtro_suavizado));

    % Asignar el valor calculado al píxel correspondiente en la imagen de salida
    OutSuavizado(rxp, ryp) = nuevo_valor;
end
end
OutSuavizado = uint8(OutSuavizado);

```

En esta sección, se crea una matriz de ceros llamada OutSuavizado para almacenar la imagen suavizada. Luego se define la máscara de suavizado filtro_suavizado, que es una matriz de 3x3 con todos sus elementos iguales a 1/9. Esto representa un filtro de promedio que suaviza la imagen al calcular el promedio de los valores de los píxeles en una vecindad de 3x3. Se utiliza un doble bucle for para recorrer la imagen, excluyendo los bordes. En cada iteración, se extrae una vecindad 3x3 del píxel actual, se multiplica elemento a elemento con la máscara, y se suma el resultado para obtener un nuevo valor que se asigna al píxel correspondiente en OutSuavizado. Finalmente, se convierte la matriz OutSuavizado a uint8 para asegurar que los valores estén en el rango correcto para visualización.

3. Filtro Box

```

OutBox = zeros(FIL, COL);
filtro_box = [0,0,0,0,0;0,1,1,1,0;0,1,1,1,0;0,1,1,1,0;0,0,0,0,0]/9;
for rxp = 3:FIL-2
    for ryp = 3:COL-2
        vecindad = double(In(rxp-2:rxp+2, ryp-2:ryp+2));
        nuevo_valor = sum(sum(vecindad .* filtro_box));
        OutBox(rxp, ryp) = nuevo_valor;
    end
end
OutBox(1, :) = 0; % Primera fila
OutBox(FIL, :) = 0; % Última fila
OutBox(:, 1) = 0; % Primera columna
OutBox(:, COL) = 0; % Última columna
OutBox = uint8(OutBox);

```

Aquí se inicializa la matriz OutBox de ceros para almacenar la imagen resultante del filtro box. La máscara filtro_box es una matriz de 5x5 con valores de 1 en el centro y 0 en los bordes, todos divididos por 9. Esto da más peso a los píxeles centrales en la vecindad,

generando un efecto de suavizado más fuerte en comparación con el filtro de promedio. Se recorre la imagen con un doble bucle for, dejando un margen de 2 píxeles en los bordes. Para cada píxel interior, se extrae una vecindad de 5x5, se multiplica con la máscara, y se suma para obtener el nuevo valor del píxel. Después de recorrer la imagen, se establecen los valores de los bordes (primeras y últimas filas y columnas) a 0 para crear un borde negro alrededor de la imagen filtrada. La matriz OutBox se convierte a uint8 para la visualización

4. Filtro Gaussiano

```
OutGauss = zeros(FIL, COL);  
sigma = 1;  
[x, y] = meshgrid(-2:2, -2:2); % Máscara de 5x5  
filtro_gaussiano = exp(-(x.^2 + y.^2) / (2 * sigma^2)); % Formula gaussiana  
filtro_gaussiano = filtro_gaussiano / sum(filtro_gaussiano(:)); % Normalizar la máscara  
for rxp = 3:FIL-2  
    for ryp = 3:COL-2  
        % Vecindad 5x5 del píxel actual (rxp, ryp)  
        vecindad = double(In(rxp-2:rxp+2, ryp-2:ryp+2));  
  
        % Aplicar la máscara gaussiana  
        nuevo_valor = sum(sum(vecindad .* filtro_gaussiano));  
  
        % Asignar el valor calculado al píxel correspondiente en la imagen de salida  
        OutGauss(rxp, ryp) = nuevo_valor;  
    end  
end  
OutGauss = uint8(OutGauss);
```

En esta sección se inicializa una matriz OutGauss para almacenar la imagen resultante del filtro gaussiano. Se define una máscara gaussiana de 5x5 utilizando la fórmula de la función gaussiana, donde se calculan los valores de la máscara en función de la distancia de cada píxel al centro y una desviación estándar (sigma) de 1. La máscara se normaliza para que la suma de sus elementos sea igual a 1, lo que asegura que la imagen resultante no cambie de brillo. Luego se recorre la imagen, excluyendo los bordes, y se extrae una vecindad de 5x5 para cada píxel. Se multiplica esta vecindad por la máscara gaussiana y se suma para obtener el nuevo valor del píxel, que se asigna a OutGauss. La matriz se convierte a uint8 para la visualización final

4. Filtro Laplaciano

```
OutLaplaciano = zeros(FIL, COL);

filtro_laplaciano = [ 0  0 -1  0  0;
                     0 -1 -2 -1  0;
                     -1 -2 16 -2 -1;
                     0 -1 -2 -1  0;
                     0  0 -1  0  0];

for rxp = 3:FIL-2
    for ryp = 3:COL-2
        % Vecindad 5x5 del píxel actual (rxp, ryp)
        vecindad = double(In(rxp-2:rxp+2, ryp-2:ryp+2));

        % Aplicar la máscara Laplaciana
        nuevo_valor = sum(sum(vecindad .* filtro_laplaciano));

        % Asignar el valor calculado al píxel correspondiente en la imagen de salida
        OutLaplaciano(rxp, ryp) = nuevo_valor;
    end
end
OutLaplaciano = uint8(OutLaplaciano);
```

En esta parte, se inicializa la matriz OutLaplaciano para almacenar la imagen resultante del filtro Laplaciano. La máscara filtro_laplaciano es una matriz de 5x5 que destaca los cambios bruscos en la imagen, lo que resalta los bordes. Se aplica de manera similar a las secciones anteriores, recorriendo la imagen con un doble bucle for, excluyendo los bordes. Para cada píxel, se extrae una vecindad de 5x5, se multiplica con la máscara Laplaciana y se suma el resultado para obtener el nuevo valor del píxel. Este valor se asigna a OutLaplaciano, y al final, la matriz se convierte a uint8 para su correcta visualización.

5. Visualización de imágenes

```
figure;
imshow(In);
title('Imagen Original');
figure;
subplot(2, 2, 1);
imshow(OutSuavizado);
title('Imagen Suavizada (Media 3x3)');
subplot(2, 2, 2);
imshow(OutBox);
```




CENTRO DE ENSEÑANZA TECNICA INDUSTRIAL

PRACTICAS DE LABORATORIO

```
title('Filtro Box con Bordes Negros');  
subplot(2, 2, 3);  
imshow(OutGauss);
```

```
title('Filtro Gaussiano 5x5 (sigma=1)');  
subplot(2, 2, 4);  
imshow(OutLaplaciano);  
title('Filtro Laplaciano 5x5');
```

Finalmente, se muestran los resultados de los filtros aplicados. Primero, se muestra la imagen original en escala de grises en una ventana independiente. Luego, en otra figura, se muestran las imágenes filtradas en una cuadrícula de 2x2 utilizando `subplot`. En la primera posición se muestra la imagen suavizada con el filtro de media 3x3, en la segunda el filtro box con bordes negros, en la tercera el filtro gaussiano de 5x5, y en la cuarta el filtro Laplaciano.

RESULTADOS Y CONCLUSIONES



CENTRO DE ENSEÑANZA TECNICA INDUSTRIAL

PRACTICAS DE LABORATORIO

REFERENCIAS

1. Scatec. (s. f.). ¿Qué es el procesamiento de imágenes? Recuperado el 27 de septiembre de 2024, de <https://www.scatec.es>
2. Universidad de Zaragoza. (s. f.). Fundamentos del procesamiento de imágenes digitales. Recuperado el 27 de septiembre de 2024, de <https://cps.unizar.es>
3. UJI (s. f.). Filtros en el procesamiento de imágenes digitales. Recuperado el 27 de septiembre de 2024, de <https://www.uji.es>
4. Centro de Tecnologías Avanzadas. (s. f.). Procesamiento de imágenes y visión artificial. Recuperado el 27 de septiembre de 2024, de <https://www.cta.com>

Elavoro	Observaciones	Evaluacion
DR. Gerardo García Gil		