

FiltradoNoLineal

Tomas Lopez Pérez

May 2024

Contents

1	Introduction	1
2	Fundamento teórico	2
2.1	Procesamiento de imágenes	2
2.2	Filtrado no lineal	3
2.3	Max y Min filtro	3
2.4	Mediana y mediana ponderada	3
2.5	Punto Medio	3
2.6	Media recortada	4
3	Metodologia	4
3.1	Definición del problema y selección de la imagen de entrada . . .	4
3.2	Implementación de los filtros	4
4	Resultados	7
5	Conclusion	8

1 Introduction

En el procesamiento de imagenes, los filtros juegan una papel crucial al mejorar mejorar la calidad visual y realizar carcateristicas de interés. Si bien los metodos de filtrado lineal han sido ampliamente utilizados, su eficacia puede verse limitada, Por lo tanto surge la necesidad de explorar y aplicar tecnicas de filtrado no lineal. En este trabajo nos centramos en la implementacion de filtros no lineales (Median Filter, MidPoint, Alpha-Trimmed Mean Filter) para imagenes en escala de grises. Estos filtros ofrecen enfoques distintos para el procesamiento de imagenes.

2 Fundamento teórico

2.1 Procesamiento de imágenes

El Procesamiento de imágenes es un campo de investigación muy extenso que involucra diversas áreas del conocimiento, ya que involucra diversos procesos, tales como: la adquisición, transmisión, representación y procesamiento. En términos generales el procesamiento de imágenes se utiliza para modificar una imagen con el fin de mejorar la apariencia visual para un observador y para resaltar convenientemente el contenido de la imagen de cara a la percepción por parte de máquinas, es decir, se hace una manipulación de imágenes con objeto de producir nuevas imágenes que son mejores, en algún sentido. El procesamiento de imágenes comprende distintas técnicas que se utilizan para mejorar, restaurar e inclusive comprimir imágenes ya existentes; la implementación y desarrollo de dichas técnicas agregadas a sistemas ya existentes nos permiten editar imágenes de forma más sencilla y precisa [Alonso2018]. La Figura 1 muestra el flujo de las etapas del procesamiento digital de imágenes propuestas por Gonzalez en donde se establece como la etapa inicial la adquisición de la imagen digital, lo que se puede hacer mediante algún sistema de sensores que permitan digitalizar la imagen u obtener las imágenes de algún repositorio o colección. El conjunto de imágenes adquiridas conforma inicialmente la base de conocimiento, ya que para algunas aplicaciones basta con tener la información pura, sin embargo en la mayoría de las aplicaciones del procesamiento de imágenes, es necesario enriquecer la base del conocimiento con al menos las etapas de procesado y segmentación, lo cual requiere aplicar una serie de operadores que transformen las imágenes para resaltar características relevantes, en el caso del procesado, o separar los diferentes elementos que componen la imagen, en el caso de la segmentación, de modo que los resultados de estas dos etapas permiten tener una base del conocimiento más amplia y robusta. Respecto a la etapa de representación y descripción. las imágenes procesadas y/o segmentadas son usadas para extraer características más específicas sobre la imagen, tales como cantidad, dimensión, forma, posición, etc. de los objetos de interés en las imágenes, esta información también puede formar parte de la base del conocimiento, de modo que en la ultima etapa, reconocimiento e interpretación, se puedan obtener respuestas a partir de las imágenes y la información generada por estas.

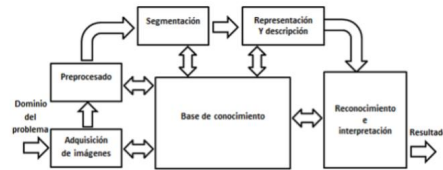


Figure 1: Etapas fundamentales del procesamiento digital de imágenes.

2.2 Filtrado no lineal

La idea general en el filtrado de imágenes no lineal es que en lugar de utilizar una máscara espacial en el proceso de convolución, la máscara es usada para obtener los valores de los píxeles vecinos.

2.3 Max y Min filtro

El filtro de máximo se utiliza para resaltar los detalles más brillantes de una imagen, como bordes o áreas de alto contraste. A diferencia del filtro de máximo, el filtro de mínimo se utiliza para suavizar la imagen y reducir la nitidez de los detalles.

$$I'(u, v) = \min_{(r, c) \in R} \{I(r, c)\}$$
$$I'(u, v) = \max_{(r, c) \in R} \{I(r, c)\}$$

Figure 2: Max and Min Filter

2.4 Mediana y mediana ponderada

Es una técnica comúnmente utilizada para eliminar el ruido impulsivo mientras se conservan los detalles importantes de la imagen.

$$I'(u, v) = \text{median}_{(r, c) \in R} \{I(r, c)\}$$

Figure 3: Median Filter

2.5 Punto Medio

El filtro de punto medio en imágenes es útil para suavizar el ruido de manera adaptativa, ya que ajusta dinámicamente su respuesta dependiendo de la variabilidad local de los valores de píxeles en la imagen.

$$I'(u, v) = \frac{1}{2} [\max_{(r, c) \in R} \{I(r, c)\} + \min_{(r, c) \in R} \{I(r, c)\}]$$

Figure 4: Midpoint filter

2.6 Media recortada

El filtro de media podada alfa es útil para reducir varios tipos de ruido, incluido el ruido gaussiano y el ruido impulsivo, mientras se conservan los detalles importantes de la imagen.

$$I'(u, v) = \frac{1}{mn - d} \sum_{(r, c) \in S_{xy}} g_R(r, c)$$

Figure 5: Midpoint filter

3 Metodologia

La metodologia propuesta en este trabajo constituye una serie de pasos para aplicar el filtrado no lineal a las imagenes, se realizo la implementacion del Median filter, Midpoint filter, Alpha-Trimmed Mean Filter, Weighted Median Filter.

3.1 Definición del problema y selección de la imagen de entrada

Tenemos como objetivo el estudio de los filtros procesamiento de la imagenes como parte del aprendizaje y la compresion de tecnicas de filtrado no lineal. Se seleccionó una imagen de prueba representativa que contenga características y estructuras diversas.

3.2 Implementación de los filtros

Para Median filter se recorre la imagen y se obtiene la mascara partiendo del pixel actual, la mascara obtenida se pasa a un vector para luego ordenarla y obtener con la funcion median de matlab la mediana y ese valor lo asignamos a la nueva imagen.

```

function imgNoFilter = MedianFilter(img)
    [width, height] = size(img);
    imgNoFilter = zeros(width-1, height-1);
    for row=2:width-1
        for col=2:height-1
            mask = [img(row-1,col-1) img(row-1,col) img(row-1,col+1);
                    img(row,col-1)   img(row,col)   img(row,col+1);
                    img(row+1,col-1) img(row+1,col) img(row+1,col+1)];
            flattening = sort(mask(:));
            imgNoFilter(row,col) = median(flattening);
        end
    end
end

```

Figure 6: Codigo Median Filter

Para MidPoint se tiene que realizar el mismo proceso de recorrer la imagen y appasar la mascara en vector y luego seleccionamos el valor minimo y el maximo para despues sumarlos y divirlos entre de dos.

```

function imgNoFilter = MidpointFilter(img)
    [width, height] = size(img);
    imgNoFilter = zeros(width-1, height-1);
    for row=2:width-1
        for col=2:height-1
            mask = [img(row-1,col-1) img(row-1,col) img(row-1,col+1);
                    img(row,col-1)   img(row,col)   img(row,col+1);
                    img(row+1,col-1) img(row+1,col) img(row+1,col+1)];
            flattening = mask(:);
            newPixel = max(flattening) + min(flattening);
            imgNoFilter(row,col) = newPixel / 2;
        end
    end
end

```

Figure 7: Codigo MidPoint

Para la implementacion de Alpha-Trimmed Mean Filter se necesecita un parametro extra que lo llamaremos $d/2$ el cual nos dice cuantos valores de cada extremos vamo a eliminar de mi mascara, entonces mi mascara la pasamos a un vector ordenamos con el metodo sort quitamos de los extremos $d/2$ y luego realizamos el promedio para que este resultado lo asignamos como una nueva intensidad de mi imagen nueva.

```

function imgNoFilter = ATMFILTER(img , d)
[width, height] = size(img);
imgNoFilter = zeros(width-1, height-1);
d = d/2;
for row=2:width-1
    for col=2:height-1
        mask = [img(row-1,col-1) img(row-1,col) img(row-1,col+1);
                img(row,col-1)   img(row,col)   img(row,col+1);
                img(row+1,col-1) img(row+1,col) img(row+1,col+1)];
        flattening = sort(mask(:));
        newFlat = flattening(d+1: 9-d);
        newPixel = sum(newFlat)/ size(newFlat,1);
        imgNoFilter(row,col) = newPixel;
    end
end
end

```

Figure 8: Codigo Alpha-Trimmed Mean Filter

Para la implementacion de Weighted Median Filter se necesita una mascara extra 10 la cual la utilizamos para repetir la intensidad de mi mascara original, una vez que tenemos el nuevo vector con los nuevos valores ordenamos y obtenemos la mediana.

```

function imgNoFilter = WMF(img , W)
[width, height] = size(img);
imgNoFilter = zeros(width-1, height-1);
[m, n] = size(W);
for row=2:width-1
    for col=2:height-1
        mask = [img(row-1,col-1) img(row-1,col) img(row-1,col+1);
                img(row,col-1)   img(row,col)   img(row,col+1);
                img(row+1,col-1) img(row+1,col) img(row+1,col+1)];

        preFlattening = mask(:);
        posW = W(:);
        repeatF = repelem(preFlattening, posW);
        sortinfF = sort(repeatF);
        imgNoFilter(row,col) = median(sortinfF);
    end
end
end

```

Figure 9: Codigo Mediana Ponderada

1	2	1
2	3	2
1	2	1

Figure 10: W

4 Resultados

La implementacion de los filtros no lineales en imagenes en escala de grises, nos da una imagen un poco borrosa en todos los resultados no se nota mucha diferencia pero si observamos los histogramas podemos ver pequeñas variaciones pero son muy sutiles.

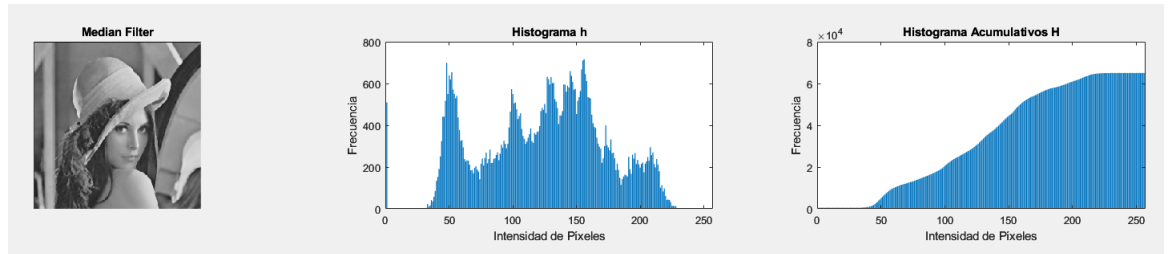


Figure 11: Codigo Median Filter

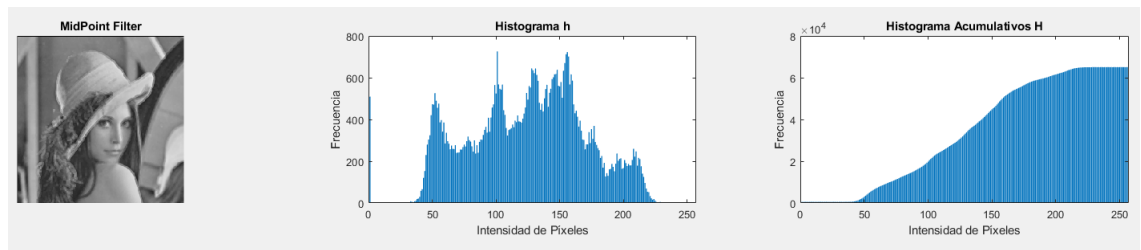


Figure 12: Codigo Punto Medio

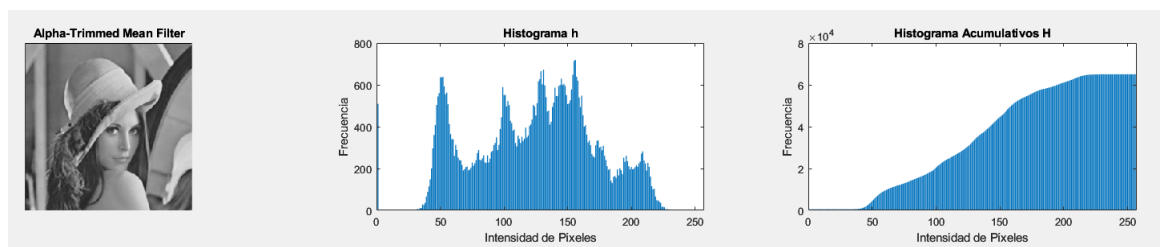


Figure 13: Codigo media recortada

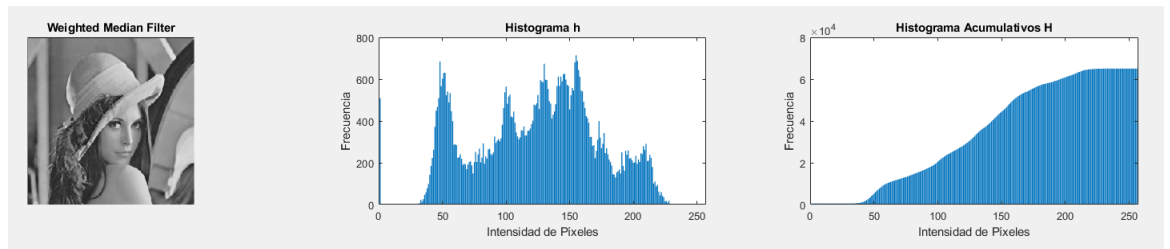


Figure 14: Código Mediana Ponderada

5 Conclusion

Se ha implementado y evaluado un conjunto de filtros para procesar imágenes en escala de grises, que incluyen el filtro de mediana, el filtro de punto medio, el filtro de media recortada (Alpha-Trimmed Mean) y el filtro de mediana ponderada (Weighted Median). Aunque los resultados en la imagen de prueba son similares, cada filtro tiene un proceso de implementación único que se adapta a diferentes necesidades y contextos de aplicación.