



INSTITUTO TECNOLÓGICO DE COSTA RICA

ÁREA DE INGENIERÍA EN COMPUTADORES

PROCESAMIENTO Y ANÁLISIS DE IMÁGENES DIGITALES

Eliminación de Ruido *Salt and Pepper* en Video

Estudiantes:

Arturo CÓRDOBA V.
Fabián GONZÁLEZ A.
Erick CARBALLO P.
Sebastián MORA R.

Profesor:

PhD. Juan Pablo SOTO

29 de junio de 2020

1. Filtro de la Mediana

El filtro de la mediana es un filtro paso bajo que trabaja en el dominio espacial, es decir, bajo el plano de la imagen original y se basa directamente en la manipulación de los píxeles de la imagen. Este es un filtro particular, ya que no se aplica la operación de convolución, por lo tanto se le conoce como un filtro no lineal y es comunmente utilizado para limpiar una imagen con ruido de tipo "Sal y Pimienta".

El filtro consiste en ordenar los valores de bloque del vecindario de un pixel, donde se calcula la mediana para encontrar el valor medio del bloque y se reemplaza el valor del pixel original con la mediana de este vecindario. Se debe destacar que el tamaño del bloque del vecindario depende de la posición del pixel. Por lo tanto, las esquinas de la imagen tiene un bloque de 4 píxeles en su vecindario incluyendo el pixel de la esquina. Los bordes superior, inferior, izquierda y derecha presenta un bloque de 6 píxeles, y por último los píxeles de la parte central de la imagen presentan un bloque de 9 píxeles de tamaño. En la figura 1 se puede observar como resultaría el vecindario o bloque para un pixel en la parte central. La figura 2 muestra el vecindario de un pixel en el borde superior y la figura 3 muestra el vecindario de una imagen en la esquina inferior derecha.

$A_{i-1,j-1}$	$A_{i-1,j}$	$A_{i-1,j+1}$
$A_{i,j-1}$	$A_{i,j}$	$A_{i,j+1}$
$A_{i+1,j-1}$	$A_{i+1,j}$	$A_{i+1,j+1}$

Figura 1: Vecindario de Pixel Central

$A_{i,j-1}$	$A_{i,j}$	$A_{i,j+1}$
$A_{i+1,j-1}$	$A_{i+1,j}$	$A_{i+1,j+1}$

Figura 2: Vecindario de Pixel en Borde

$A_{i-1,j-1}$	$A_{i-1,j}$
$A_{i,j-1}$	$A_{i,j}$

Figura 3: Vecindario de Pixel en Esquina

1.1. Pseudocódigo

El siguiente pseudocódigo muestra el proceso de filtrado solamente para la parte central de la imagen, en la cual presenta un bloque o vecindario de 9 pixeles, que resultaría en una matriz de tamaño 3×3 . En primer lugar se calcula cual es el tamaño de la imagen original. Se realiza un recorrido de la parte central de la imagen, donde para cada pixel se obtiene su bloque o vecindario. Este bloque se almacena en una variable de forma vectorizada para luego calcular con un comando propio de GNU Octave la mediana de este vector, que se reemplaza el valor del pixel por el resultado de esta mediana. Para las esquinas y los bordes se realizaría un proceso similar, pero con un ventana de diferente tamaño

Algorithm 1 Filtro de la Mediana

Entradas: A : Imagen Original

Salida: B : Imagen Filtrada

```
1:  $[m, n] = \text{size}(A)$ 
2: for  $x = 2, \dots, m - 1$  do
3:   for  $y = 2, \dots, n - 1$  do
4:      $\text{bloque} = A(x - 1 : x + 1, y - 1 : y + 1)$ ;
5:      $\text{vecAux} = \text{bloque}(:)$ ;
6:      $B(x, y) = \text{median}(\text{vecAux})$ 
7:   end for
8: end for
9: return  $B$ 
```

1.2. Resultados

Se implementó computacionalmente en GNU Octave este filtro de la mediana para el eliminar el ruido de tipo "Sal y Pimienta" del video con nombre `video-con-ruido.mp4` y se obtiene como resultado un video a color con nombre `video-sin-ruido-1.mp4`. El proceso de filtrado presentó una duración en tiempo de aproximadamente 18.583 minutos. En la figura 4 y 6 se muestran imágenes del video con ruido de tipo "Sal y Pimienta", mientras que en las figuras 5 y 7 se muestran las imágenes con el filtro de la mediana aplicado.



Figura 4: Imagen del Video con Ruido



Figura 5: Imagen Filtrada del Video (Filtro de la Mediana)



Figura 6: Imagen del Video con Ruido

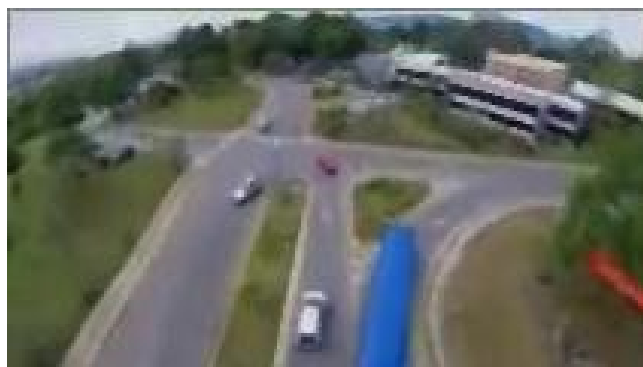


Figura 7: Imagen Filtrada del Video (Filtro de la Mediana)

2. Algoritmo IAMFA-I

El algoritmo IAMFA-I propuesto en artículo [1] es un algoritmo diseñado para mejorar la imagen de salida para una aproximación del filtro de la mediana rápido implementado por Marcus y Ward en su artículo [2].

El algoritmo implementa una técnica llamada "Mediana de la Decisión del Valor Medio" (Mid-Value-Decision Median en inglés) que minimiza las posibilidades de seleccionar un valor corrupto para reemplazar el valor del pixel original de una ventana. La técnica selecciona el valor de la mediana de una columna si y solo si no es considerado un pixel con ruido.

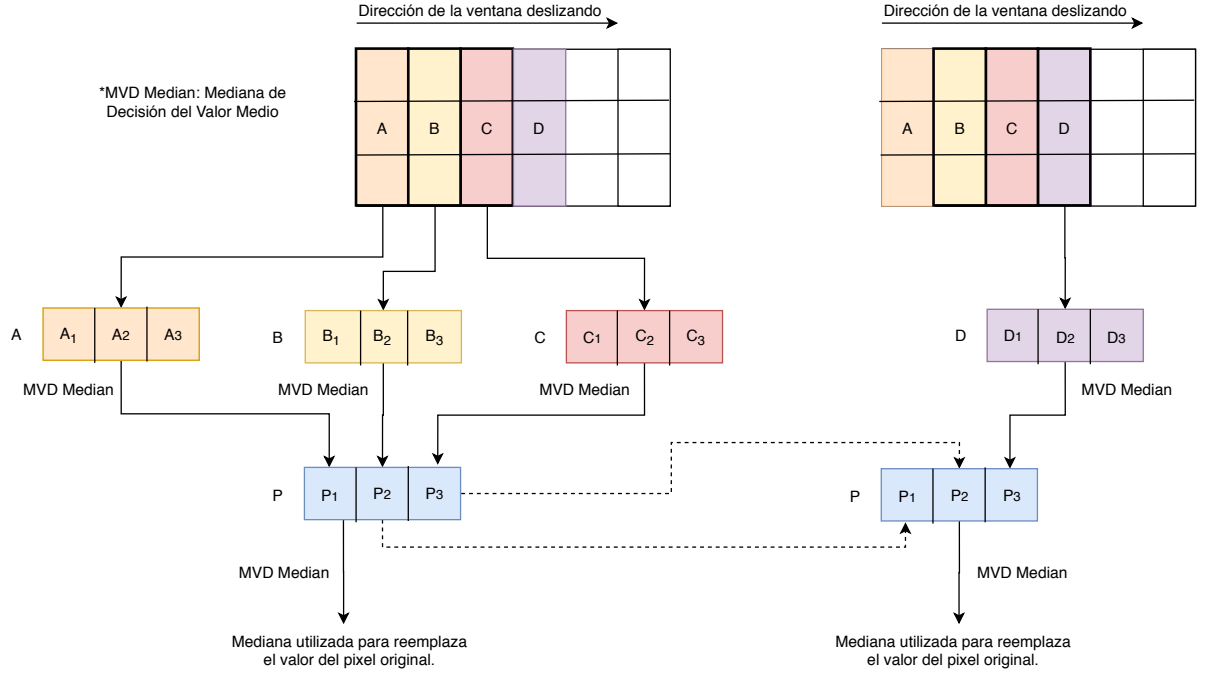


Figura 8: Ilustración del Algoritmo IAMFA-I

Por ejemplo en la figura 8, se asume que A almacena una columna extraída de la ventana. Cuando A es ordenado de forma ascendente por su valores, se puede representar como $A = \{A_1 \leq A_2 \leq A_3\}$. El valor que se seleccionará para el reemplazo del píxel original de esta ventana dependerá del valor actual de A_2 . La mediana de la decisión del valor medio se selecciona de la siguiente manera:

$$\text{Mid-Value-Decision Median} = \begin{cases} A_1 & \text{si } A_2 = 255 \\ A_2 & \text{si } 0 < A_2 < 255 \\ A_3 & \text{si } A_2 = 0 \end{cases} \quad (1)$$

2.1. Pseudocódigo

El siguiente pseudocódigo muestra el proceso de filtrado solamente para la parte central de la imagen, en la cual presentan ventanas de 9 pixeles, que resultaría en una matriz de tamaño 3×3 . En primer lugar se calcula cual es el tamaño de la imagen original. Se realiza un recorrido de la parte central de la imagen, donde para cada una de las columnas de la ventana se calcula la mediana de la decisión del valor medio. Se calcula una vez más la mediana de la decisión del valor medio para el vector que contiene los resultados de las medianas de cada columna, donde este resultado reemplaza el valor del pixel original. Para las esquinas y los bordes de la imagen se utiliza el filtro de la mediana visto en el problema anterior.

Algorithm 2 Filtro de la Mediana

Entradas: A : Imagen Original

Salida: B : Imagen Filtrada

```
1:  $[m, n] = \text{size}(A)$ 
2: for  $i = 2, \dots, m - 1$  do
3:    $col1 = \text{Mid-Value-Decision Median}([A(i - 1, 1) \ A(i, 1) \ A(i + 1, 1)]);$ 
4:    $col2 = \text{Mid-Value-Decision Median}([A(i - 1, 2) \ A(i, 2) \ A(i + 1, 2)]);$ 
5:   for  $j = 3, \dots, n - 1$  do
6:      $col3 = \text{Mid-Value-Decision Median}([A(i - 1, j) \ A(i, j) \ A(i + 1, j)]);$ 
7:      $B(i, j) = \text{Mid-Value-Decision Median}([col1 \ col2 \ col3]);$ 
8:      $col1 = col2;$ 
9:      $col2 = col3;$ 
10:  end for
11: end for
12: return  $B$ 
```

2.2. Resultados

Se implementó computacionalmente en GNU Octave el algoritmo de IAMFA-I para el eliminar el ruido de tipo "Sal y Pimienta" del video con nombre `video-con-ruido.mp4` y se obtiene como resultado un video a color con nombre `video-sin-ruido-2.mp4`. El proceso de filtrado presentó una duración en tiempo de aproximadamente 15.548 minutos. En la figura 9 y 11 se muestran imágenes del video con ruido de tipo "Sal y Pimienta", mientras que en las figuras 5 y 7 se muestran las imágenes con el algoritmo IAMFA-I aplicado.



Figura 9: Imagen del Video con Ruido



Figura 10: Imagen Filtrada del Video (Algoritmo IAMFA-I)



Figura 11: Imagen del Video con Ruido

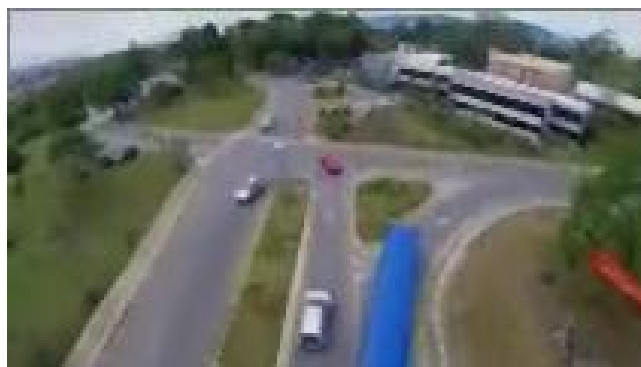


Figura 12: Imagen Filtrada del Video (Algoritmo IAMFA-I)

3. Índice de Similitud Estructural (SSIM)

El índice de similitud estructural (SSIM sus siglas en inglés) es un método para determinar la similitud entre dos imágenes. El índice SSIM resultante entre dos imágenes es un valor decimal en el intervalo $[0,1]$, donde el valor de 1 solo es accesible en el caso de que las dos imágenes sea idénticas, lo cual indicaría una similitud estructural perfecta. En caso contrario, un valor de 0 indica que no hay similitud estructural entre las imágenes.

Se utilizó una función de nombre *ssim* para determinar cual de los dos métodos, ya sea el filtro de la mediana o el algoritmo IAMFA-I realiza una mejor limpieza o filtrado del video con ruido. En la tabla 1 se muestran los resultados el índice de similitud estructural para ambos métodos. Se puede observar que el filtro de la mediana presenta un SSIM más cercano a 1 que en comparación al algoritmo IAMFA-I, por lo tanto el método del filtro de la mediana realiza una mejor limpieza del video con ruido. Esto se debe a que el filtro de la mediana realiza el cálculo de la mediana con vecindario muchos más amplios que el algoritmo IAMFA-I, por lo tanto se selecciona un valor de pixel más ajustable. También el algoritmo IAMFA-I, a pesar de que se trata de obtener una mejor imagen de salida, está modificando un filtro de la mediana con una aproximación más rápida, lo cual quiere decir que no siempre se quiere obtener un buen filtrado, sino que el proceso de filtrado sea más rápido.

Método	SSIM
Filtro de la Mediana	0.82488
Algoritmo IAMFA-I	0.76279

Tabla 1: Resultados del SSIM

Referencias

- [1] O. Appiah, M. Asante, and J.B. Hayfron-Acquah, "Improved approximated median filter algorithm for real-time computer vision applications", Journal of King Saud University - Computer and Information Sciences, April 2020.
- [2] R. Marcus, W. Ward, "DP: a Fast Median Filter Approximation", pp 1-11, 2013.