

TAREA #1 — Procesamiento y Análisis de Imágenes Digitales

1. Variaciones del Filtro de la Mediana

El filtro de la mediana es uno de los filtros predominantes en el procesamiento digital de imágenes para reducir el ruido impulsivo o ruido "sal y pimienta". No obstante, su algoritmo estándar suele presentar tiempos de ejecución significativos y/o bajo desempeño cuando la imagen está altamente corrompida. Lo anterior ha llevado a desarrollar variaciones del algoritmo, algunas enfocadas en reducir el tiempo de ejecución del filtro y otras en generar salidas de calidad [1]. A continuación se verán dos algoritmos, el Fast Median Filter Approximation (DP) e IAMFA-I, ambos son variaciones del filtro de la mediana diseñadas para mejorar la velocidad de procesamiento y la calidad de los resultados al filtrar ruido en imágenes.

1.1. Algoritmo 1: DP - Fast Median Filter Approximation

Es un algoritmo no discreto que aproxima rápidamente un filtro de mediana, fue propuesto por Marcus y Ward (2013). A diferencia del filtro de la mediana estándar, que requiere ordenar todos los valores de la ventana para encontrar la mediana, el DP aprovecha la superposición de datos debido al deslizamiento de ventana durante el proceso de filtrado. En cada iteración se trabaja con una ventana de 3 x 3, donde se toma la mediana de cada columna y entre esas tres medianas se toma su mediana [1]. El principal desafío de DP es que el algoritmo puede reemplazar un píxel no corrompido con uno corrompido, lo que lleva a salidas deficientes generadas después de la operación de eliminación de ruido. La Figura 1 ilustra cómo se procesa una ventana y cómo se desliza a través de la imagen desde el borde izquierdo hasta el derecho.

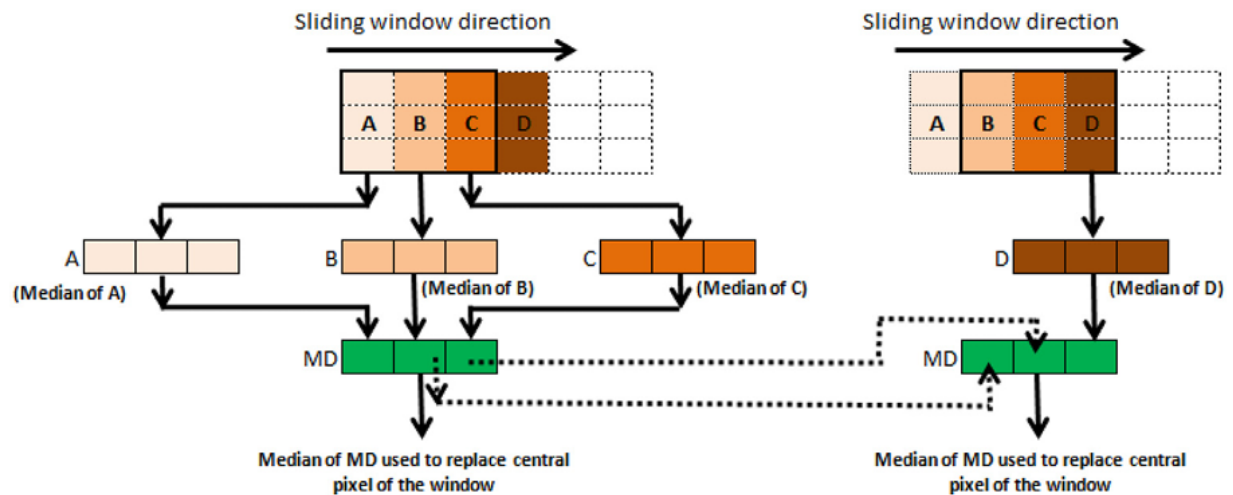


Figura 1: Ilustración de DP: Fast Median Filter Approximation [1]

Algorithm 1: DP – Fast Median Filter Approximation

```
1 let  $I$  represent the image to be filtered;
2 let  $H$  be the HEIGHT of the Image;
3 let  $W$  be the WIDTH of the Image;
4 for  $i = 2$  to  $H - 1$  do
5    $col1 \leftarrow \text{median}(I(i - 1, 1), I(i, 1), I(i + 1, 1));$ 
6    $col2 \leftarrow \text{median}(I(i - 1, 2), I(i, 2), I(i + 1, 2));$ 
7   for  $j = 3$  to  $W - 1$  do
8      $col3 \leftarrow \text{median}(I(i - 1, j), I(i, j), I(i + 1, j));$ 
9      $I'(i, j) \leftarrow \text{median}([col1; col2; col3]);$ 
10     $col1 \leftarrow col2;$ 
11     $col2 \leftarrow col3;$ 
```

1.2. Algoritmo 2: Proposed Improved Approximated Algorithm - IAMFA-I

Este algoritmo tiene un tiempo de ejecución equivalente a DP, pero mejora su salida (aún cuando aumenta el ruido impulsivo de una imagen). El algoritmo introduce una técnica llamada Mid-Value-Decision-Median (MVDM), la cual busca minimizar la probabilidad de seleccionar un píxel con ruido como el valor central de la ventana. La MVDM se asegura de que el valor seleccionado sea el más representativo (el que tiene menos probabilidades de estar afectado por el ruido) dentro de los tres valores que se comparan.

Suponiendo que P almacena una partición o una columna extraída de una ventana. Cuando P se ordena en orden ascendente, el arreglo se puede definir como $P = \{P_1 \leq P_2 \leq P_3\}$. El valor a seleccionar para el reemplazo del valor central de una ventana dependerá del valor actual de P_2 . La técnica tiene una mayor probabilidad de seleccionar un píxel que no sea de impulso para el filtrado en comparación con el método original utilizado en el DP [1]. El MVDM corresponde a:

$$\text{Mid - Value - Decision Median} = \begin{cases} P_1, & \text{si } P_2 = 255 \\ P_2, & \text{si } 0 < P_2 < 255 \\ P_3, & \text{si } P_2 = 0 \end{cases}$$

Algorithm 2: Proposed Improved Approximated Algorithm - IAMFA-I

```
1 let  $I$  represent the image to be filtered;
2 let  $H$  be the HEIGHT of the Image;
3 let  $W$  be the WIDTH of the Image;
4 for  $i = 2$  to  $H - 1$  do
5    $col1 \leftarrow \text{MVDM}(I(i - 1, 1), I(i, 1), I(i + 1, 1));$ 
6    $col2 \leftarrow \text{MVDM}(I(i - 1, 2), I(i, 2), I(i + 1, 2));$ 
7   for  $j = 3$  to  $W - 1$  do
8      $col3 \leftarrow \text{MVDM}(I(i - 1, j), I(i, j), I(i + 1, j));$ 
9      $I'(i, j) \leftarrow \text{MVDM}([col1; col2; col3]);$ 
10     $col1 \leftarrow col2;$ 
11     $col2 \leftarrow col3;$ 
```

2. Eliminación de Ruido en un Video

En esta sección se grabó un video con una duración de 7 segundos y una resolución de 320x240 píxeles. A dicho video se le aplicó ruido de tipo salt and pepper para posteriormente usar los algoritmos 1 y 2 con el objetivo de filtrar el ruido y obtener una salida similar al video original. En las figuras 2, 3, 4 y 5 se muestran el primer frame del video original, con ruido, sin ruido debido al algoritmo 1 y sin ruido debido al algoritmo 2.



Figura 2: Primer frame del video original.

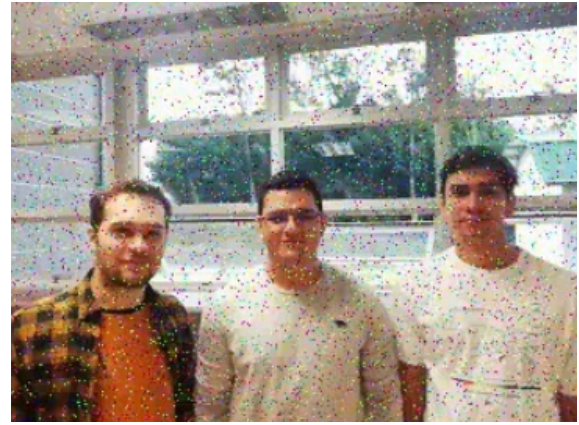


Figura 3: Primer frame del video con ruido.



Figura 4: Primer frame del video sin ruido con el algoritmo 1.



Figura 5: Primer frame del video sin ruido con el algoritmo 2.

Utilizando ssim para determinar cuál de los dos métodos o algoritmos realiza una mejor limpieza del video con ruido de tipo salt and pepper, se observa según los resultados que para este caso en particular y la cantidad de ruido que se le agregó al video original, que ambos algoritmos logran una similitud media de reconstrucción o filtro similar. No obstante, cabe destacar que algoritmo 2 tiene un tiempo de ejecución menor que el del algoritmo 1.

```
Command Window
>> resultado_ssim

SSIM promedio con Algoritmo 1:
0.7668
SSIM promedio con Algoritmo 2:
0.7600
>> |
```

Figura 6: Resultado del SSIM promedio con algoritmo 1 y 2 en comparación con el video original.

3. Referencias Bibliográficas

1. Appiah, Obed and Asante, Michael and Hayfron-Acquah, James Benjamin. *Improved approximated median filter algorithm for real-time computer vision applications*. Elsevier, 2022. Disponible en: <https://www.sciencedirect.com/science/article/pii/S1319157820303402>

Submitted by Ignacio Grané Rojas - Emanuel Antonio Marín Gutiérrez - Luis Felipe Vargas Jimenez on 25 de Agosto, 2024.