

## **Procesamiento y Análisis de Imágenes Digitales**

### **Tarea #2**

#### **Parte 1**

#### **Estudiantes:**

Daniel Castro Campos - 2016165624

Joseph Vargas Blanco - 2016200170

Gustavo Segura Umaña - 2016089824

**Docente: Luis Chavarría Zamora**

**II Semestre 2021**

# Parte 1. Eliminación de Ruido del Tipo Salt and Pepper en un vídeo

## Algoritmos.

### Algoritmo Fast Median Filter Approximation

Este algoritmo es una implementación aproximada del filtro de la mediana, reduciendo el tiempo de computación respecto al algoritmo original. En esta implementación se utiliza una ventana 3x3 que se divide en tres columnas y la mediana de cada columna es calculada. La mediana de las medianas luego se estima y se toma este valor para reemplazar el pixel central de la ventana.

```
I = imagen por filtrar
H, W = alto y ancho de la imagen

for i = 2, i++, H - 1
    columna1 = mediana(I(i-1, 1), I(i, 1), I(i+1, 1))
    columna2 = mediana(I(i-1, 2), I(i, 2), I(i+1, 2))

    for j = 3, j++, W - 1
        columna3 = mediana(I(i-1, j), I(i, j), I(i+1, j))
        I_filtrado(i, j) = mediana([columna1; columna2; columna3])
        columna1 = columna2
        columna2 = columna3
    endfor
endfor
```

*Pseudocódigo del algoritmo*

## Algoritmo HPDBMF

El algoritmo HPDBMF o High-Performance Modified Decision Based Median Filter Algorithm, es otro algoritmo para aproximar el filtro de la mediana en una imagen. Este es una modificación del algoritmo Modified Decision Based Median Filter. en el cual también se toma una ventana 3x3 en la que se utiliza como el nuevo pixel la mediana de todos los valores de píxeles de esa ventana 3x3 a excepción de los que tengan valor 0 o 255.

En el caso de que todos los valores sean 0 o 255, se amplía la ventana a 5x5 y se repite el procedimiento. Si de nuevo, todos los valores son 0 o 255, se utiliza el valor del pixel calculado anteriormente.

```
I = imagen
for pixel in imagen
    if 0 < Pij < 255, usar Pij para la imagen filtrada
    else
        Seleccionar ventana 3x3 con el pixel en el centro
        Posicionar la ventana en tres columnas (col1, col2, col3)
        M1 = MVDM(col1_1, col2_1, col3_1)
        M2 = MVDM(col1_2, col2_2, col3_2)
        M3 = MVDM(col1_3, col2_3, col3_3)

        Mediana = MVDM(M1, M2, M3)

        Si el valor de Mediana es invalido, incrementar ventana a
        5x5 y seleccionar el primer encuentro de un pixel no-impulso como valor
        de Mediana
        Si Mediana sigue siendo inválido, seleccionar el último
        pixel procesado como el valor
        Utilizar Mediana para la imagen filtrada
    end
endfor

function MVDM(X, Y, Z)
    Ordenar X, Y, Z en orden ascendente, asignarlos a P1, P2 y P3
    respectivamente
    if P2 es 0, entonces retornar P3
    elseif P2 es 255, entonces retornar P1
    else retornar P2
end
```

*Pseudocódigo del algoritmo*

## Índice de Similitud.

El índice de similitud se encarga de determinar la semejanza entre dos imágenes por lo que es posible utilizarlo para realizar una comparación entre los videos filtrados y el original. Un 1 indica semejanza perfecta mientras que un 0 indica que es completamente diferente, al aplicar este índice a los videos generados por los filtros se obtienen los siguientes resultados:

SSIM comparando entre los videos originales por lo que debe dar 1:

**1**

SSIM comparando video original contra filtrado por el Filtro de Mediana Rápida:

**0.9344**

SSIM comparando video original contra filtrado por HPDBMF:

**0.9755**

Es posible observar que el mejor filtrado lo realiza el algoritmo HPDBMF y esto tiene una explicación muy lógica al comparar el funcionamiento de los dos filtros, este filtro se encarga de modificar únicamente los valores impulsivos sean 0 o 255, por lo que ataca directamente las irregularidades causadas por el ruido, mientras que el filtro de mediana rápida hace un filtrado de la imagen completa.