Tarea 6 - Filtros Salt n' Pepper y mediana en Simulink

Asignatura: Procesamiento de imágenes
Profesor: D. Sc. Gerardo García Gil
Alumno: Jonathan Guillermo Díaz Magallanes.
Registro: 19310153. **Ciclo:** 2022-B
Ingeniería en Desarrollo de Software
Centro de Enseñanza Técnica Industrial (CETI)

Introducción

En el transcurso de ésta práctica se desarrollarán dos filtros; *Salt n' Pepper* y *mediana*. Primero se realizará el *Salt n' Pepper* para añadir ruido a la imagen en cuestión y después el filtro de mediana para eliminar el ruido producido. Ésto se realizará por medio del lenguaje científico MATLAB, además de implementarlos en Simulink.

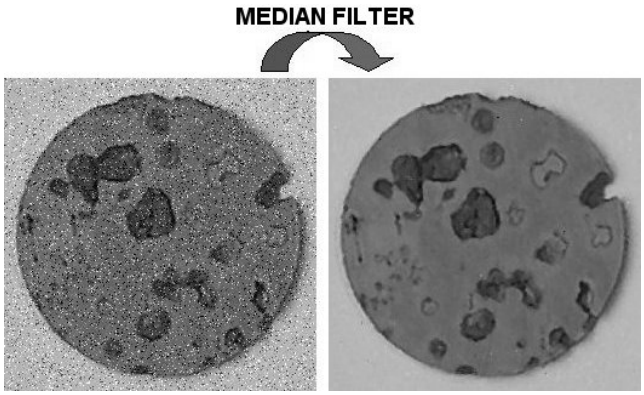
Filtro Salt n' Pepper

El ruido sal y pimienta (salt-and-pepper noise) que se presenta principalmente en imágenes. Se caracteriza principalmente por cubrir de forma dispersa toda la imagen con una serie de píxeles blancos y negros.

Este tipo de ruido suele producirse cuando la señal de la imagen es afectada por intensas y repentinas perturbaciones o impulsos.

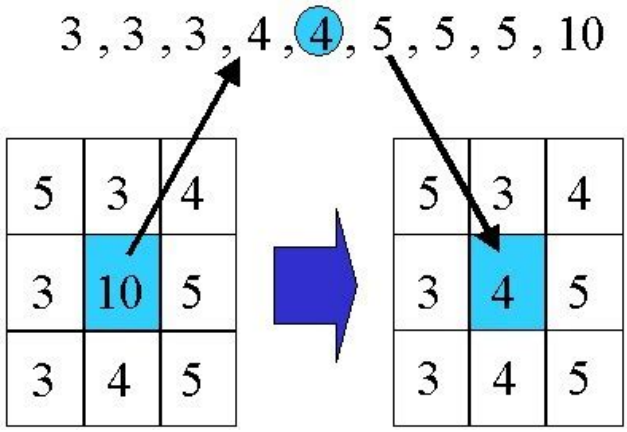
Filtro mediana

El filtro mediano es una técnica de filtrado digital no lineal, a menudo utilizada para eliminar el ruido de una imagen o señal. Dicha reducción de ruido es un paso típico de preprocesamiento para mejorar los resultados del procesamiento posterior. El filtrado mediano es muy utilizado en el procesamiento digital de imágenes porque, bajo ciertas condiciones, preserva los bordes mientras elimina el ruido, también tiene aplicaciones en el procesamiento de señales.



La idea principal del filtro mediano es correr a través de la señal entrada por entrada, reemplazando cada entrada con la mediana de las entradas vecinas. El patrón de vecinos se llama "ventana", que se desliza, entrada por entrada, sobre toda la señal. Para las señales unidimensionales, la ventana más obvia son solo las primeras entradas anteriores y siguientes, mientras que para los datos bidimensionales la

ventana debe incluir todas las entradas dentro de un radio o región elipsoidal dada.

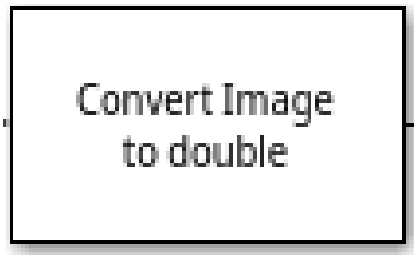


Desarrollo

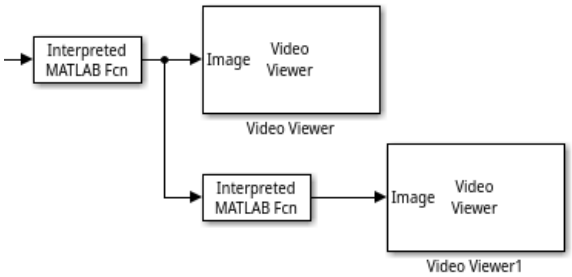
Lo primero que se realizó fué el diagrama de bloques de Simulink.

El primer bloque que se usó es el que se encarga de obtener la imagen de video proporcionada por la webcam de mi laptop. Éste bloque se encuentra dentro del paquete *Image acquisition toolbox*, el bloque se llama *from video device*:

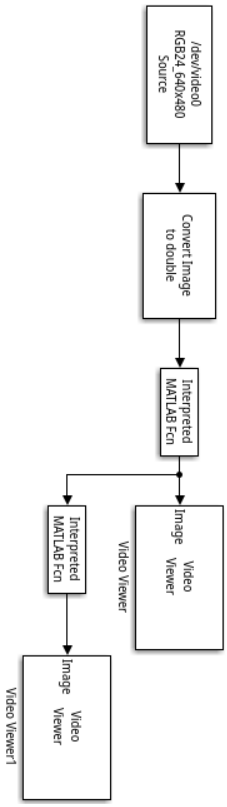
El bloque lo vamos a configurar para que obtenga la imagen de la webcam en RGB y en la salida nos dé la misma imagen directamente en escala de grises.
La salida de éste bloque la vamos a conectar a otro bloque el cual nos va a convertir ésta imagen en valores double.



Como salida obtendremos a la imagen en formato double la cual está lista para ingresarla al interpretador de funciones de MATLAB, el cual es nuestro siguiente bloque en el diagrama de bloques. Se utilizarán dos interpretadores, uno para aplicar el filtro *Salt n' Pepper* y otro para el filtro *mediana*. Cabe mencionar que a la entrada del interpretador que contenga la función del filtro *mediana* se le proporcionará la salida del interpretador con la función del filtro *Salt n' Pepper*. Además a cada uno de los interpretadores se les conectará un *Video viewer* a sus salidas para poder visualizar el efecto de los filtros.

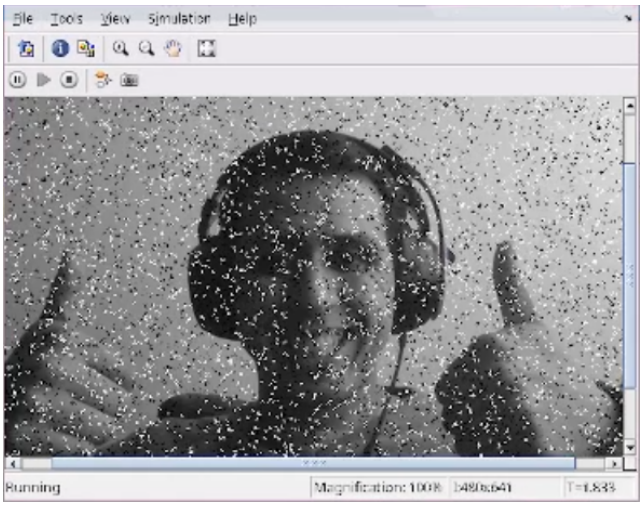


Al final con todos los bloques conectados nos quedaría un diagrama así como el siguiente:

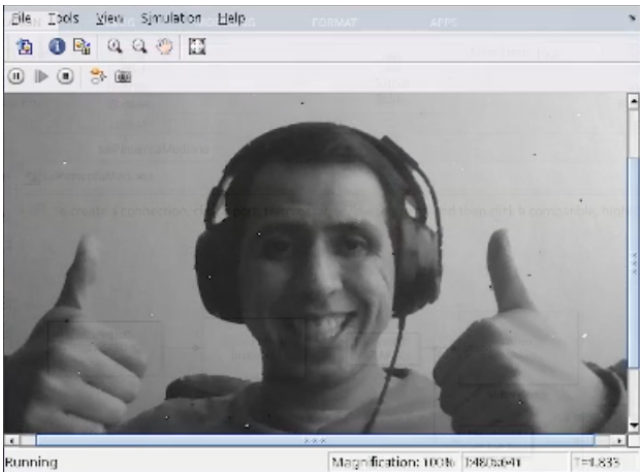


Para la ejecución del programa se le asignó un ruido de 5000 partículas al filtro *Salt n' Pepper*. Al momento de ejecutar el programa se obtuvieron los siguientes resultados:

Filtro Salt n' Pepper



Filtro mediana



Código

Filtro Salt n' Pepper

```
function Dr = ruidoSM(I)

Intensidad = 5000;

[Filas, Columnas, P] = size(I);

Dr = I;

% Empezamos con el filtro de ruido sal y pimienta

% Puntos blancos

for v=1:Intensidad

 x=round(rand*Filas);

 y=round(rand*Columnas);

 if (x==0) || (x==1) || (x==2)

  x=3;

 end

 if (y==0) || (y==1)|| (y==2)

  y=3;

 end

 if (x==Filas) || (x==Filas-1) || (x==Filas-2)

  x=Filas-3;

 end

 if (y==Filas) || (y==Filas-1) || (y==Filas-2)

  y=Columnas-3;

 end

 for i=1:4

  rRandom = randi([-1 1]);

  cRandom = randi([-1 1]);

  Dr(x+rRandom,y+cRandom) = 255;

 end

end

% Puntos negros

for v=1:Intensidad

 x=round(rand*Filas);

 y=round(rand*Columnas);
```

```
if (x==0) || (x==1) || (x==2)
  x=3;
end
if (y==0) || (y==1)|| (y==2)
  y=3;
end
if (x==Filas) || (x==Filas-1) || (x==Filas-2)
  x=Filas-3;
end
if (y==Filas) || (y==Filas-1) || (y==Filas-2)
  y=Columnas-3;
end
for i=1:4
  rRandom = randi([-1 1]);
  cRandom = randi([-1 1]);
  Dr(x+rRandom,y+cRandom) = 0;
 end
end
% Im = uint8(Dr);
Im = Dr;
end
```

Filtro mediana

```
function Im = medianaSM(I)
[Filas, Columnas, P] = size(I);
B = I;
for i=2:Filas-1
  for j=2:Columnas-1
      mat=[B(i-1,j-1), B(i-1,j), B(i-1,j+1), B(i,j-1), B(i,j), B(i,j+1), B(i+1,j-1), B(i+1,j), B(i+1,j+1)];
% Empezamos con el metodo burbuja
    for x=1:8
     for val=1:8
      if mat(val) > mat(val+1)
        temp = mat(val);
        mat(val) = mat(val+1);
        mat(val+1) = temp;
       end
     end
    end
    B(i,j)=mat(5);
  end
end
Im = B;
end
```

Conclusión

En el desarrollo de ésta práctica no se presentaron problemas mayores al momento de hacer la elaboración de cada uno de los filtros. Se tuvo problema no con la práctica en sí, sino con el programa Simulink como tal debido a que no funcionó recién instalado en linux, por ello se investigó y encontró la solución la cual radica con el motor gráfico con el que se ejecuta MATLAB. Fuera de ello no se presentó complicación o duda alguna.

Referencias

- Filtro de Mediana. (n.d.). FIRST Robotics Competition Documentation. Retrieved September 25, 2022, from https://docs.wpilib.org/es/stable/docs/software/advanced-controls/filters/median-filter.html

- Ingraham, A. K. (2018, November 28). Salt & Pepper Noise and Median Filters, Part I – The Theory. Kyle Ingraham | Engineer. Retrieved September 25, 2022, from https://blog.kyleingraham.com/2016/12/14/salt-pepper-noise-and-median-filters-part-i-the-theory/

- Remove Salt and Pepper Noise from Images - MATLAB & Simulink. (n.d.). Retrieved September 25, 2022, from https://www.mathworks.com/help/vision/ug/remove-salt-and-pepper-noise-from-images.html

- Wikipedia contributors. (2021, September 10). Median filter. Wikipedia. Retrieved September 25, 2022, from https://en.wikipedia.org/wiki/Median_filter