|  |
| --- |
| Licenciatura en Sistema |
| TP SIMD |
| Organización del Computador II |

|  |
| --- |
| Abregu, Pablo – Yagusz, Salvador – Santillán, Javier  04-06-2020 |

1. Filtro Aclarar

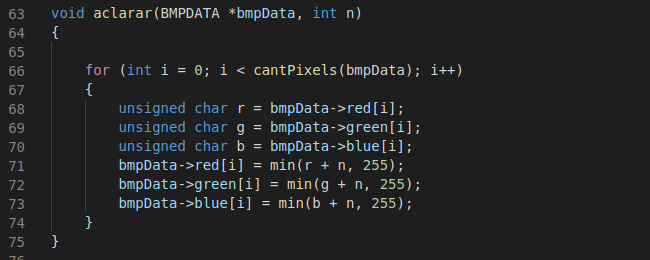
Se aplica el filtro para aclarar una imagen.

Para este filtro le suma una determinada cantidad a cada canal del pixel, siendo RGB los valores R(Red), G(Green), B(Blue):

**RGB = (R + n, G + n, B + n)**

La función recibe como parámetro una imagen a aclarar, y un valor n, siendo este el nivel de aclaración.

Código:



Prueba:

aclarar (&bmpData, 50);

Original Modificada

1. Median Filter

Se aplica el filtro de mediana.

Este consiste en aplacar el ruido de una imagen, reparando cada pixel.

En este caso se agarra una ventana de 3x3, que va recorriendo toda la imagen. Esos 9 pixeles, obtenidos en la ventana, se ordenan por cada canal del pixel RGB, dividiéndolo en 3 listas uno para cada canal del pixel.

typedef struct

{

    unsigned char r;

    unsigned char g;

    unsigned char b;

} Pixel;

typedef Pixel t\_Window[9];

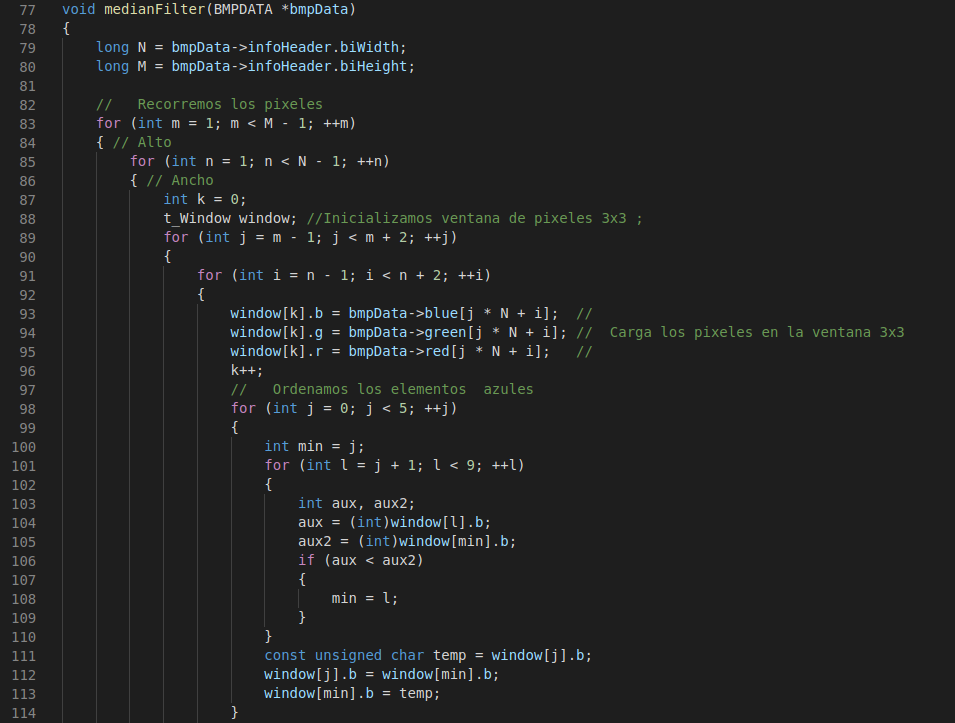
Luego se procede a ordenar cada lista de canales. y se obtiene la media del ordenamiento de la ventana. Por último, al pixel inicial de la ventana seleccionada sin ordenar se le asignan los valores de mediana de cada uno de los tres canales.

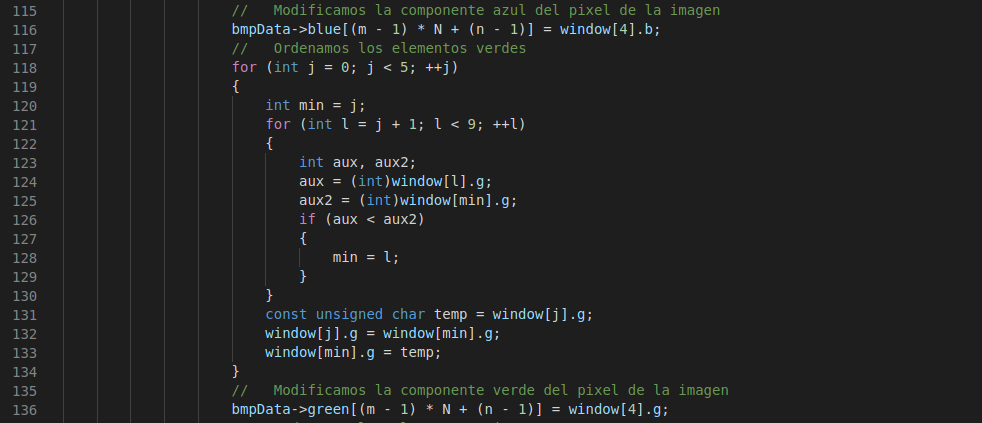
bmpData->red[(m - 1) \* N + (n - 1)]  = window[4].r;

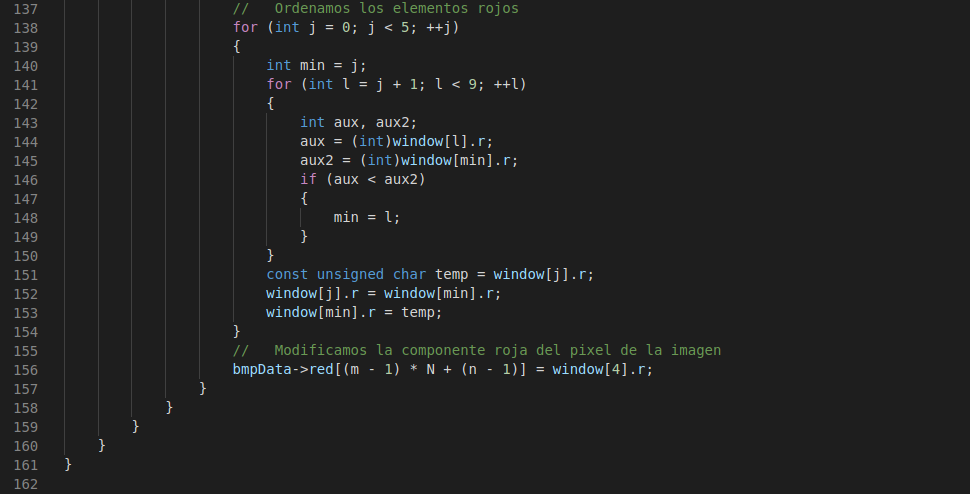
bmpData->green[(m - 1) \* N + (n - 1)] = window[4].g;

bmpData->blue[(m - 1) \* N + (n - 1)]  = window[4].b;

Codigo:







Prueba:

Imagen Corrompida Imagen Sin Ruido

1. Blend

Se aplica el filtro Blend.

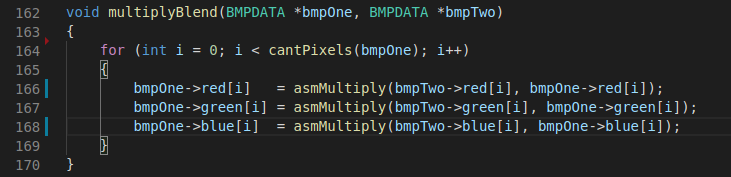
Este filtro consiste en superponer dos imágenes, multiplicando los canales de cada pixel de una imagen con la otra, generando una nueva.

La multiplicación del los valores del pixel se hace mediante código assembler (multiply.s)

**P1(r, g, b) \* P2(r, g, b) = P3(r1\*r2, g1\*g2, b1\*b2)**

Codigo:

La función recibe dos imágenes para superponer.



Multply.s

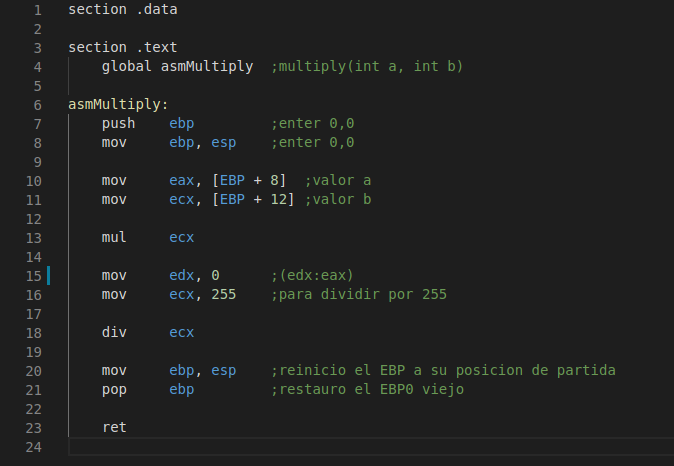


Imagen A Imagen B

Imagen generada



1. Blend MMX

Se aplica el filtro Blend.

Este filtro consiste en superponer dos imágenes, multiplicando los canales de cada pixel de una imagen con la otra, generando una nueva.

A diferencia del anterior filtro, este funciona con SIMD, utilizando instrucciones MMX.

Lo cual, aumentaría la eficacia en el tiempo de ejecución.

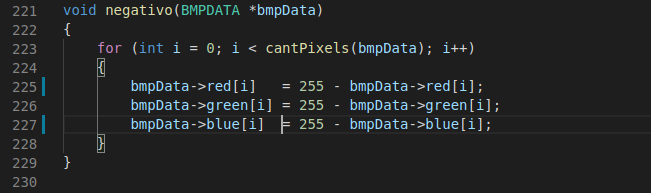
1. Filtro Negativo

Se aplica el filtro Negativo.

Este filtro consiste en la resta de 255 para cada uno de los canales del pixel.

**P1(r, g, b) = P (255 – r, 255 – g, 255 – b)**

Codigo:



Prueba:

Original Negativo

1. Filtro Escala de Grises

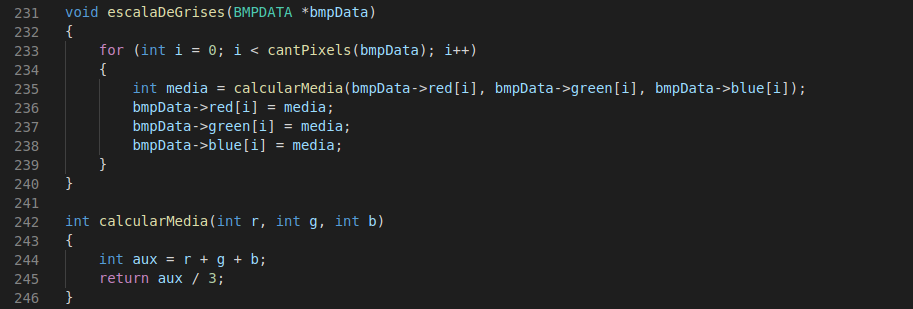
Se aplica el filtro de Escala de Grises.

Este filtro consiste en calcular la suma de cada uno de los canales del pixel, y luego dividiéndolo por 3.

**P (r, g, b) = P ((**r+g+b) / 3, (r+g+b) / 3, (r+g+b) / 3**)**

Codigo:

La función recibe una imagen por parámetro, y en cada del pixel se reemplaza por la suman de cada uno de los canales, dividiéndolo por 3.



Prueba:

Original Modificada

** **

**Repositorio en Git:** <https://github.com/pabloabregu/SIMD.git>