Examen Práctico
Cortes Cirett David

Introducción:

El procesamiento digital de imágenes tiene como objetivo principal aplicar a las imágenes ciertas técnicas que nos permiten mejorar su calidad, su aspecto, o facilitar el reconocimiento de objetos o patrones dentro de una imagen digital, extrayendo datos y convirtiéndolos en información.

Desarrollo:

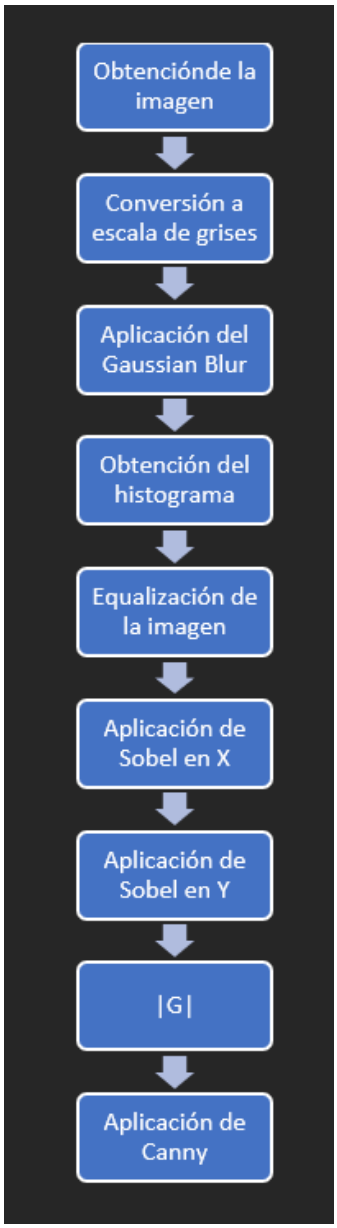


Figura 1. Diagrama de bloques.

El primer paso en el procesamiento de la imagen es su obtención, en este caso se hará uso de la imagen de llamada Lena, obtenida en una revista Playboy del año de 1972, esta imagen se encuentra, aunque

limitada por la época, a color, por lo que el siguiente paso es una conversión a escala de grises.

Conversión a escala de grises:

En esta etapa se tiene que realizar una conversión a escala de grises, se puede hacer mediante dos formas, promedio o NTSC, la cual usa la fórmula que se ve en la figura 2.

```
Grayscale = 0.299R + 0.587G + 0.114B
```

Figura 2. NTSC

En la práctica de conversión a escala de grises se hizo una pequeña comparación de ambos métodos de conversión a grises, en la cual NTSC resultó ser mejor teniendo una mayor calidad en cuanto a la fidelidad de la luminosidad. Con esta fórmula lo que se tiene que hacer es obtener el valor a color de cada píxel, una vez obtenido es multiplicado por una constante, esto lo hacemos por medio de Vec3b, que lo que hace o contiene es 3 vectores uchar de 8 bits donde obtendremos valores de 0 a 255, usada para guardar los valores del color.
Obtenemos los valores del píxel en que se está, después se multiplica y por último se tiene que checar si estamos en los valores fuera de la imagen, ya sea de lado izquierdo, derecho, arriba o abajo, esto para mostrar el borde del kernel.

Aplicación del filtro Gaussiano:

Lo siguiente de obtener la imagen en grises es aplicar el filtro gaussiano, para esto se crea un kernel dinámico, normalmente se puede hacer con apuntadores, sin embargo std cuenta con un método llamado Vector, igualmente OpenCv pero este guardará únicamente valores flotantes y no será un vector con 3 valores sino que será un vector representando a un array que en realidad es un kernel dinámico, la ventaja de vector es que no hay necesidad de liberar memoria, aunque es un poco más complejo trabajar con vectores, sin embargo cumplen la misma función que realizar un apuntador. Una vez creado el kernel lo tenemos que inicializar con la fórmula mostrada en la figura 3.

$$\frac{1}{2\pi\sigma^2}\exp\{-\frac{x^2+y^2}{2\sigma^2}\}$$

Figura 3. Filtro Gaussiano

Con esta fórmula se tiene que llenar nuestra matriz en cada punto de x,y. Al ser dinámico tenemos que hacer que el usuario meta los valores que él/ella desee, con estos valores será rellenado nuestro kernel, una vez creado correctamente, se tiene que mostrar y normalizar, esto último se tiene que hacer para poder trabajar de forma correcta con la imagen.
Finalmente con nuestra imagen en grises y el kernel creado, se procede a aplicarlo, esto no es más que multiplicar los valores de nuestro

pixel con los valores del kernel y con eso realizamos una sumatoria de valores , y en la posición de nuestro píxel central se colocará el nuevo valor de la sumatoria.

Obtención del histograma:

Con el filtro Gaussiano ya aplicado se obtiene el histograma, para ellos se tiene una matriz, en esa matriz se van a guardar cuantas veces tenemos un valor de gris en nuestra imagen. obteniendo la frecuencia de cada una de los píxeles de la imagen.

Ecualización de la imagen:

Una vez obtenidos e indexados los valores se procede a ecualizar la imagen, finalmente se tiene que realizar el promedio de los valores entre el tamaño de la imagen, con eso se realiza una sumatoria para los valores ecualizados, con eso obtenemos nuestra imagen del filtro de gauss ecualizada. Con esto realizado se procede a introducir los valores en una imagen de salida.

Aplicación de Sobel en X:

Con una imagen ecualizada es más fácil de detectar bordes, para aplicar el filtro de sobel se tiene un kernel ya predefinido de 3x3 para x, sin embargo se puede trabajar con valores más grandes, para aplicar este filtro se tiene que iterar primero en la matriz de la imagen, y también mientras se itera esa parte, hay que iterar en la parte del kernel, por ejemplo en la posición (i,j) = (0,0) se está iterando el kernel moviéndose del centro al rededor con los vecinos. Al igual que en Gauss se realiza una sumatoria de la multiplicación de los valores como se muestra en la figura 4.

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}$$

Figura 4. Sobel en X

Estos valores se guardan en una nueva imagen de muestra.

Aplicación de Sobel en Y:

Con una imagen ecualizada es más fácil de detectar bordes, para aplicar el filtro de sobel se tiene un kernel ya predefinido de 3x3 para x, sin embargo se puede trabajar con valores más grandes, para aplicar este filtro se tiene que iterar primero en la matriz de la imagen, y también mientras se itera esa parte, hay que iterar en la parte del kernel, por ejemplo en la posición (i,j) = (0,0) se está iterando el kernel moviéndose del centro al rededor con los vecinos. Al igual que en Gauss se realiza una sumatoria de la multiplicación de los valores como se muestra en la figura 5.

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

Figura 5. Sobel en Y

Estos valores se guardan en una nueva imagen de muestra.

|G|:

Con estos valores obtenidos, se procede a realizar la magnitud o normalización de los valores esto no es más que la sumatoria de los valores de cada pixel de x,y al cuadrado como se muestra en la figura 6.

$$\mathbf{G} = \sqrt{\mathbf{G}_x^{\,2} + \mathbf{G}_y^{\,2}}$$

Figura 6 Magnitud de Sobel

Este valor es introducido a una nueva imagen.

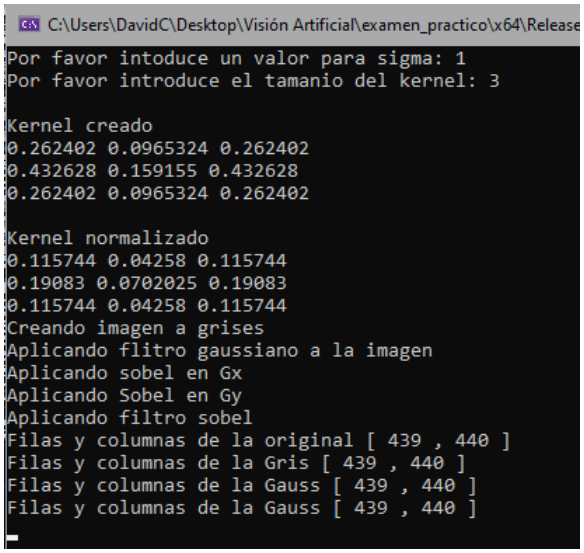
Aplicación de Canny edge detection:

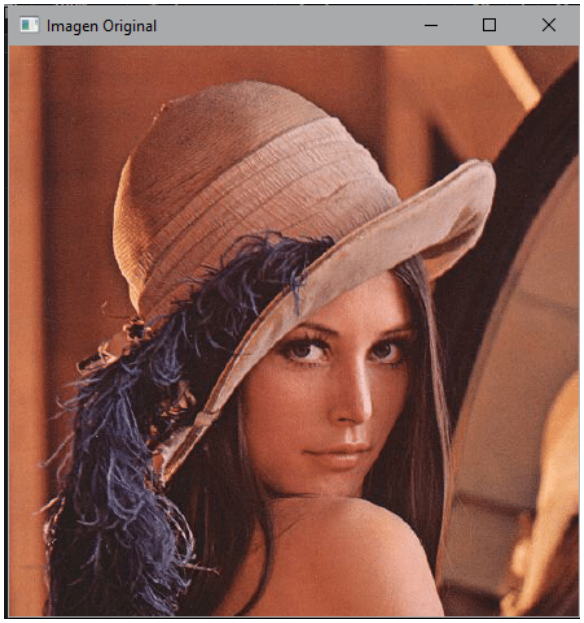
Finalmente se procede a aplicar el filtro, sinceramente se desconoce mucho de lo que se hace en este filtro, ya que nunca se vio, al igual que Sobel, solo fue mencionado y ya, así que se mencionará su procedimiento.

Lo primero que se tiene que hacer es ampliar el filtro gaussiano para remover ruido, lo segundo es obtener la intensidad del gradiente, lo tercero es aplicar la supresión no máxima para poder descartar las respuestas no esenciales, y finalmente se aplica un threshold para determinar los posibles bordes, como se muestra en la figura 7.

Figura 7. Canny Edge detector

Resultados:





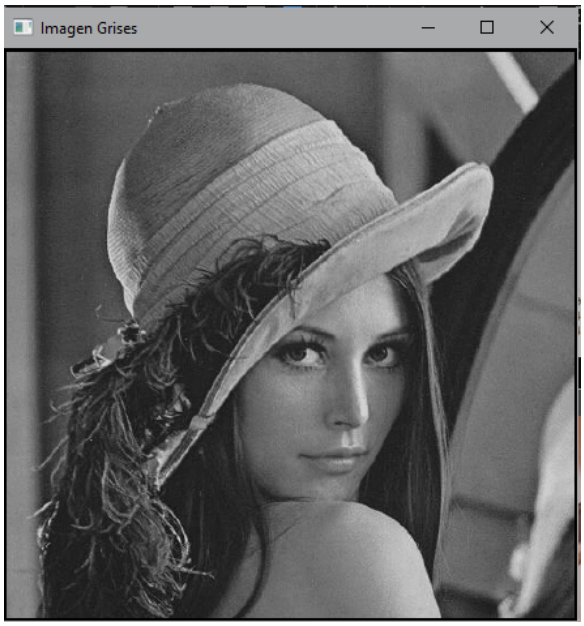


Imagen Grises

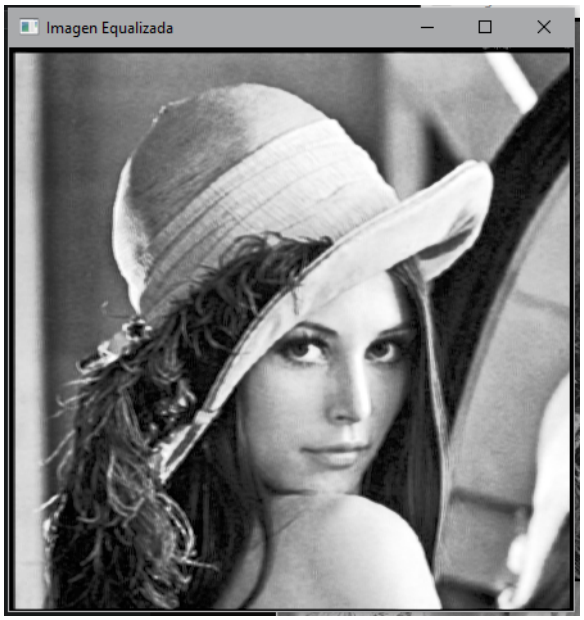


Imagen Equalizada


Imagen Blur Gaussiano


Personalizar y controlar Opera


Imagen Sobel Gy

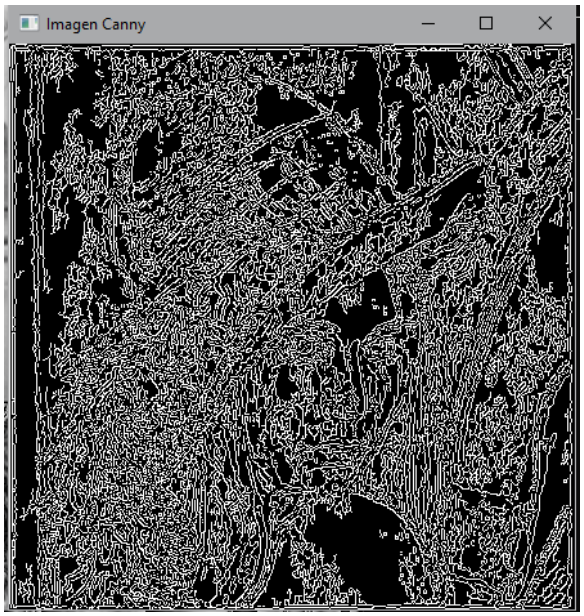


Imagen Canny

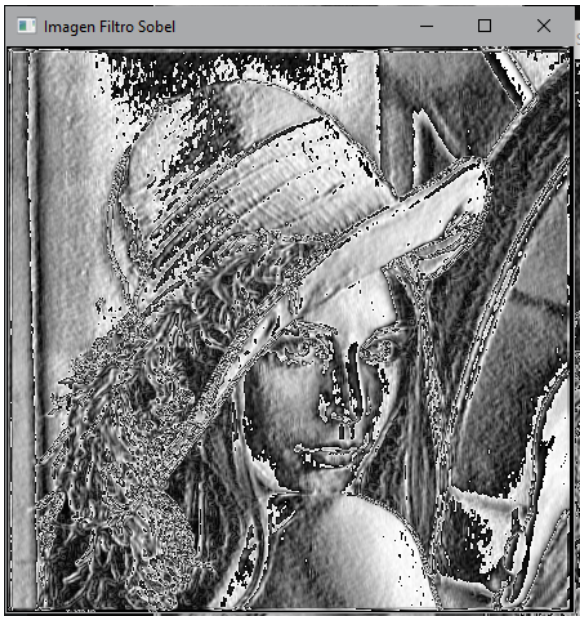


Imagen Filtro Sobel