

INSTITUTO POLITECNICO NACIONAL

Escuela Superior de Computo

Nombre: Estrada Crespo Karla Julieta

Grupo:5BM1

Materia: Visión Artificial

Examen primer parcial

**Diagrama a bloques**

Obtenemos las matrices de Gx y Gy

Volvemos a ampliar para aplicar sobel

Convertir a escala de grises

Creación del kernel

Creamos imagen ampliada

Abrir imagen

Aplicamos el filtro Gaussiano con el kernel

Ecualizamos la imagen con Filtro Gaussiano

Creamos las mascaras para sobel (Gx, Gy)

Aplicamos proceso de supresión no- máxima

Obtenemos los ángulos

Obtenemos la matriz de |G|

Aplicamos doble umbralizado

Aplicamos proceso de histeresis

Imagen final con detección de bordes Canny

Descripción del diagrama a bloques

**Abrir imagen ->** En este proceso únicamente se abre la imagen original con la que vamos a estar trabajando

**Convertir a escala de grises ->** La imagen original la pasamos a escala de grises para trabajar solo con un canal de color y que los procesos no sean tan complejos

**Creación del kernel ->** El tamaño del kernel es dinámico, es decir que las filas y las columnas serán definidas por el usuario, luego que se tiene el tamaño de la matriz se procede a obtener las coordenadas de ese kernel, posteriormente con estas coordenadas se calcularan los valores del kernel Gaussiano, todo esto mediante el uso del la formula del filtro Gaussiano.

**Creamos imagen ampliada ->** Para obtener la imagen ampliada deberemos obtener primero el tamaño de la ampliación, este se obtendrá operando el tamaño de nuestro kernel, una vez que se tenga esto sumaremos esa ampliación en todo el contorno de nuestra imagen original, de esta manera se obtendrá la imagen ampliada y esta tendrá un contorno de 0’s.

**Aplicamos el filtro Gaussiano con el Kernel ->** El kernel Gaussiano que obtuvimos con anterioridad lo pasaremos por la imagen ampliada, el kernel nos servirá como mascara para calcular el valor de cada píxel mediante la multiplicación de cada término de la máscara con los términos que empaten con estos de la imagen ampliada, una vez que se tienen todas las multiplicaciones se suma y esta suma será el nuevo valor del píxel. Para suavizar esta imagen dividiremos la suma final entre la suma de todos los valores de nuestro kernel.

**Ecualizamos la imagen con filtro Gaussiano ->** La ecualización de la imagen será realizada mediante una función de openCV que ecualiza el histograma.

**Creamos las máscaras para sobel (Gx y Gy) ->** Se crean ambas matrices para las mascaras de Gx y Gy y las llenamos con los valores establecidos.

**Volvemos a ampliar para aplicar sobel ->** Ampliamos la imagen con filtro de Gauss en una unidad, esto nos servirá para posteriormente encontrar todos los vecinos de los pixeles que se encuentran en las esquinas de nuestra imagen.

**Obtenemos las matrices de Gx y Gy ->** Pasaremos las mascaras de Gx y Gy obtenidos anteriormente sobre la imagen ampliada con filtro de Gauss y nuevamente se harán las multiplicaciones pixel a pixel para obtener una suma que será el nuevo valor del pixel que será agregado a la matriz de Gx o Gy según sea el caso.

**Obtenemos la matriz de |G| ->** Esta matriz se obtendrá mediante la formula de magnitud, que es la raíz cuadrada de la suma de las potencias cuadradas de los valores correspondientes de Gx y Gy.

**Obtenemos los ángulos ->** Para obtener los ángulos se usará la fórmula de arctan de valor correspondiente de Gy / Gx

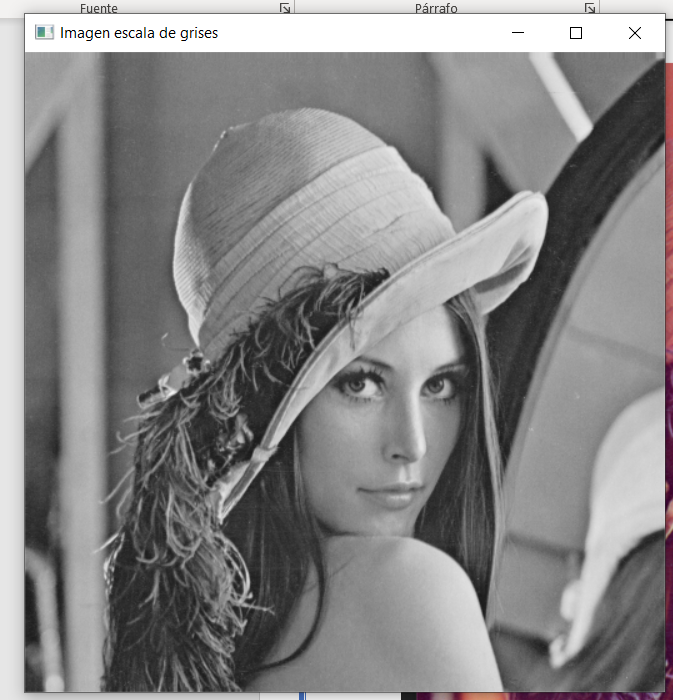
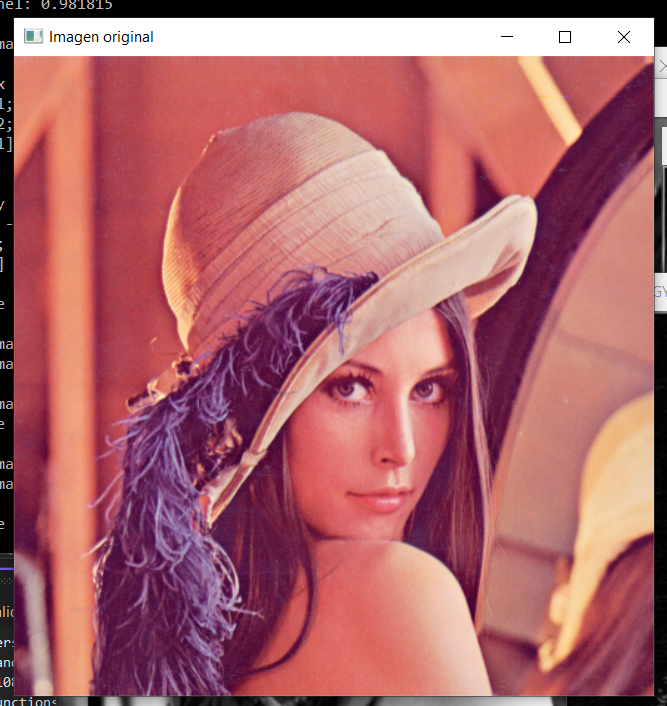
**Aplicamos proceso de supresión no-máxima ->** En este proceso se obtendrán los vecinos de los pixeles dependiendo su ángulo, para ello se declararon intervalos para tomar en cuenta los vecinos en diagonales. El algoritmo pasa por todos los pixeles de la matriz y encuentra los pixeles con el valor máximo en las direcciones de los bordes. El propósito de esto es verificar si los pixeles en la misma dirección son mas o menos intensos que los que se procesan. Si hay pixeles vecinos con mayor intensidad que el que se esta procesando, este se coloca en 0 y se toma el píxel con mayor intensidad, si no hay un píxel con mayor intensidad se conserva el que se está procesando.

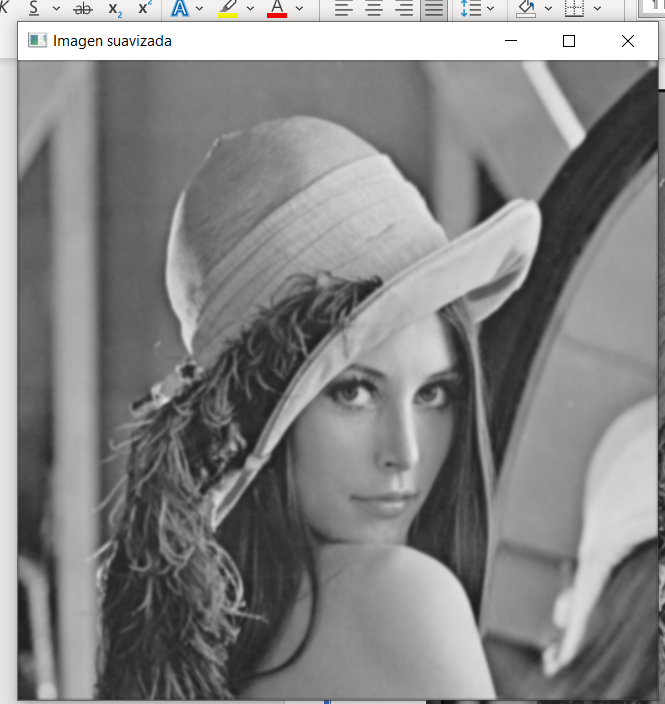
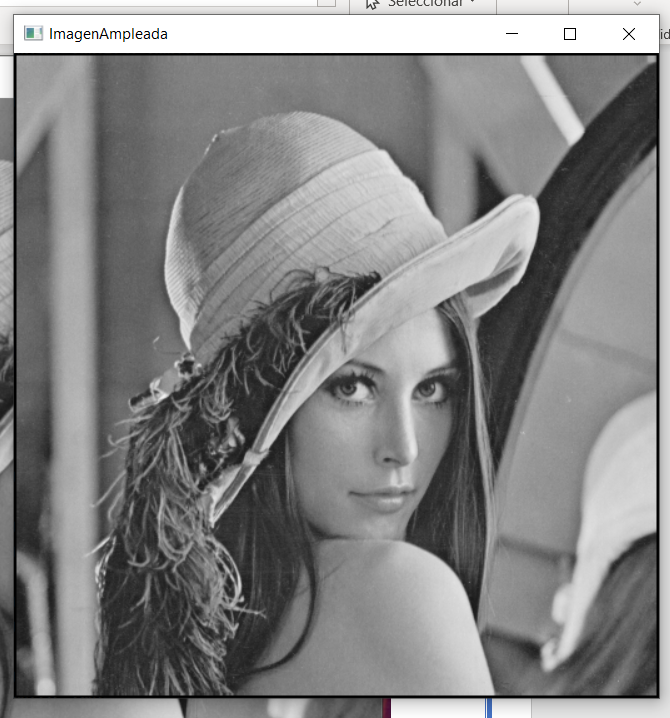
**Aplicamos doble umbralizado ->** En este proceso se tiene como objetivo identificar 3 tipos de pixeles, fuertes, débiles y no relevantes. Para ello se define un umbral alto y uno bajo, con el umbral alto se identifican los pixeles de intensidad fuerte, con el umbral bajo se identifican los pixeles no relevantes, todos los pixeles que tienen intensidades entre ambos umbrales se marcan como débiles.

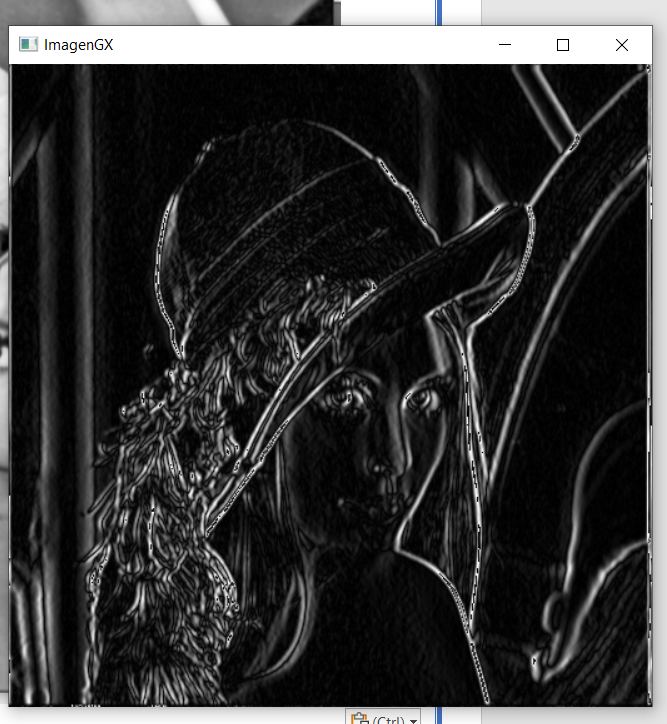
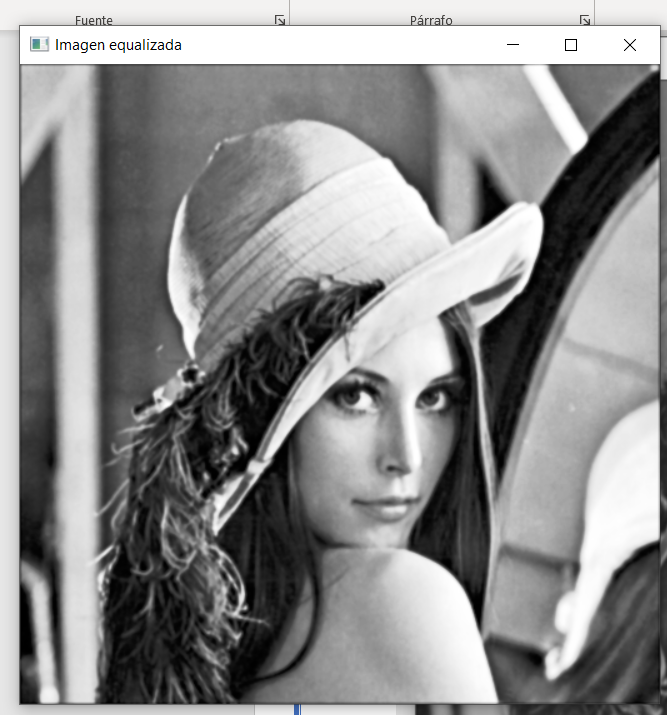
**Aplicamos proceso de histéresis ->** El proceso de histéresis consiste en transformar los pixeles débiles en fuertes si y solo si al menos uno de los pixeles alrededor del que se está procesando es fuerte.

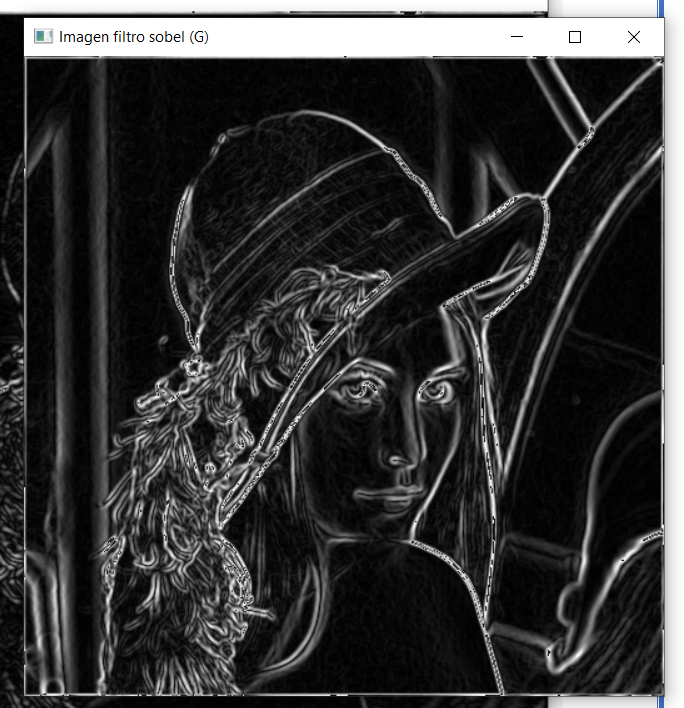
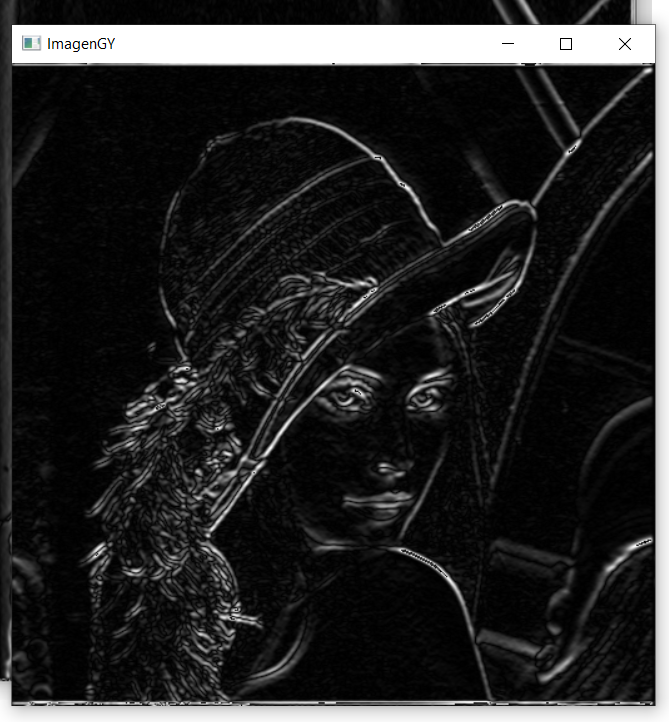
**Imagen final con detección de bordes Canny ->** Finalmente después de aplicar los filtros y procesos antes descritos se obtiene la imagen final con detección de bordes Canny.

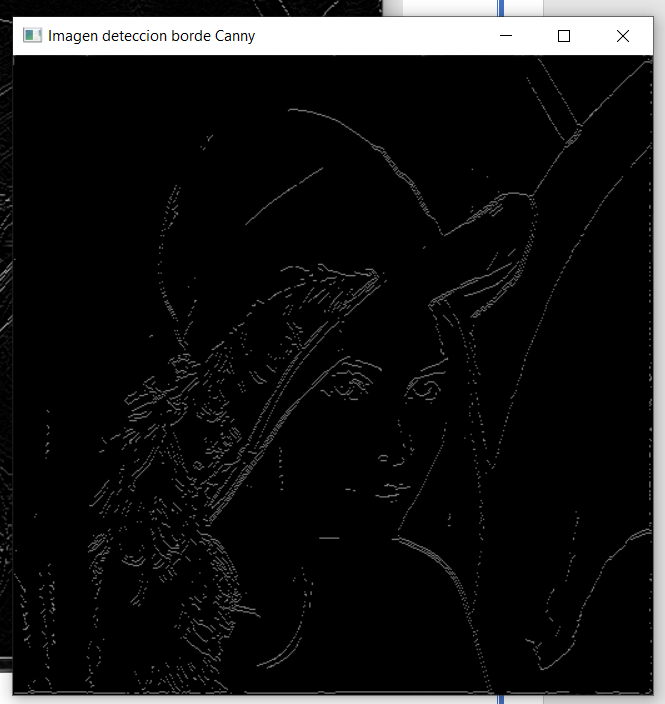
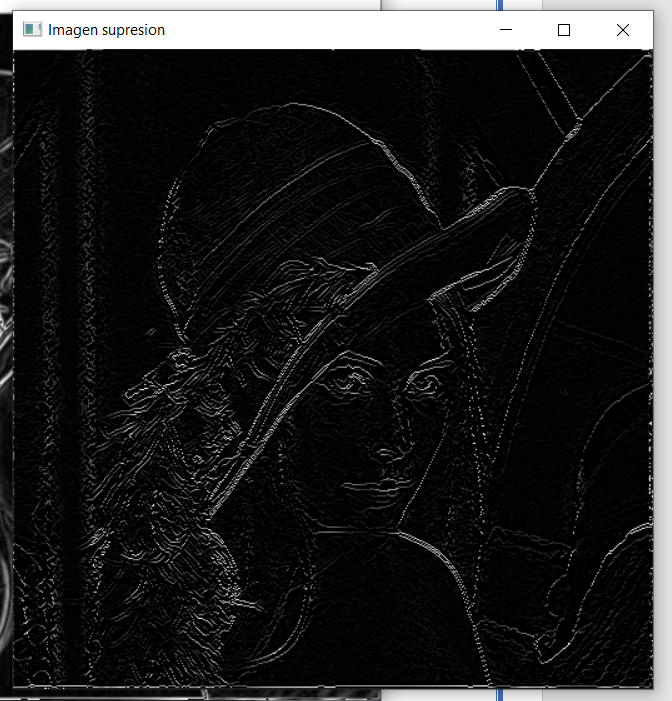
**Imágenes de transición**

****

****

****

****

****