



## **PRACTICA 2**

<b>Título:</b>	<b>Detección de bordes de Canny</b>
<b>Fecha:</b>	03-03-2023
<b>Preparado por:</b>	<b>Martínez Marqueda Luis Eduardo</b>
<b>Aprobado por:</b>	<b>Dr. Marco Antonio Negrete Villanueva</b>



## Resumen

En esta práctica, se llevó a cabo la implementación de un algoritmo de detección de bordes utilizando técnicas de filtrado y operadores de gradiente. El objetivo principal consistió en detectar y resaltar los bordes presentes en las imágenes adquiridas en tiempo real mediante una cámara.

El proceso de detección de bordes se dividió en varios pasos. En primer lugar, se aplicó un filtro Gaussiano para suavizar el video capturado por la cámara y reducir el ruido presente en la imagen. A partir del video filtrado en tiempo real, se calcularon los gradientes en los ejes “x” e “y” utilizando los operadores de Sobel. Estos gradientes fueron utilizados para obtener la magnitud y el ángulo del gradiente en cada punto de la imagen.

Con el objetivo de resaltar los bordes, se aplicó la técnica de supresión de no máximo. Esta técnica se basa en la comparación de la magnitud del gradiente con la magnitud de sus vecinos en la dirección del ángulo del gradiente. Aquellos puntos que representaron máximos locales en la dirección del gradiente fueron considerados como bordes potenciales.

Posteriormente, se realizó un umbral doble en los bordes potenciales con el fin de distinguir entre bordes fuertes y débiles. Los bordes fuertes fueron preservados como bordes finales, mientras que los bordes débiles se sometieron a un proceso de rastreo de bordes por histéresis. En este proceso, los bordes débiles fueron considerados como bordes finales si estaban conectados a bordes fuertes.

Finalmente, se mostraron las imágenes resultantes de cada etapa del proceso de detección de bordes. Esto incluyó la visualización del fotograma original, la imagen filtrada, los gradientes en las direcciones x e y, así como los bordes detectados.

## Introducción

El algoritmo de detección de bordes de Canny, propuesto por John Canny en 1986, ha ganado reconocimiento y se ha convertido en uno de los métodos más utilizados y ampliamente aceptados en el campo del procesamiento de imágenes. Desde su introducción, ha sido ampliamente estudiado y aplicado en una amplia gama de aplicaciones en diversas disciplinas. En el contexto histórico, el desarrollo del algoritmo de Canny se produjo en un momento en el que el procesamiento de imágenes estaba experimentando avances significativos. La



necesidad de detectar y extraer características importantes de las imágenes, como bordes y contornos, impulsó la investigación en algoritmos de detección de bordes más precisos y robustos.

Antes de la propuesta de Canny, existían varios métodos para detectar bordes en imágenes, pero a menudo presentaban limitaciones en términos de precisión y respuesta en presencia de ruido o discontinuidades. El algoritmo de Canny abordó estas limitaciones al incorporar técnicas como el suavizado con filtro Gaussiano, la supresión de no máximo y el umbral doble, lo que permitió mejorar significativamente la calidad y confiabilidad de la detección de bordes.

La contribución de Canny fue reconocida por su enfoque completo y su capacidad para superar los desafíos comunes en la detección de bordes, lo que llevó a una amplia adopción y reconocimiento de su algoritmo en la comunidad científica y en aplicaciones prácticas. Desde entonces, el algoritmo de Canny se ha convertido en un estándar de referencia y ha sentado las bases para muchos otros métodos y técnicas en el campo de la detección de bordes.

El contexto histórico del algoritmo de Canny destaca su relevancia y contribución en un momento en el que el procesamiento de imágenes estaba evolucionando rápidamente. Su enfoque innovador y su éxito en la mejora de la precisión y robustez de la detección de bordes han contribuido a su popularidad y a su continuo uso en numerosas aplicaciones en la actualidad.

## Objetivos

- Aprender a utilizar las funciones de convolución de OpenCV
- Aplicar el filtro de Sobel para cálculo de gradientes
- Implementar el detector de bordes de Canny

## Marco teórico

En el campo del procesamiento de imágenes, la detección de bordes es una técnica fundamental que permite identificar y resaltar los límites entre diferentes regiones en una imagen. La detección de bordes desempeña un papel crucial en numerosas aplicaciones, como reconocimiento de objetos, segmentación de imágenes, análisis de formas y mucho más.



En esta práctica, se implementa el algoritmo de detección de bordes de Canny, el cual es ampliamente utilizado debido a su precisión y capacidad para suprimir el ruido. A continuación, se describen los conceptos teóricos principales involucrados en el algoritmo de Canny, incluyendo los kernels de Gauss, los operadores de Sobel y la supresión de no máximos.

- **Filtro Gaussiano:** El filtro Gaussiano es un filtro suavizante utilizado para eliminar el ruido presente en una imagen. Aplica una convolución ponderada utilizando una función Gaussiana como kernel. El filtro Gaussiano se caracteriza por su capacidad para preservar las características de una imagen mientras reduce el ruido de alta frecuencia. En el algoritmo de Canny, el filtro Gaussiano se aplica inicialmente para suavizar el video capturado por la cámara y reducir el ruido antes de la detección de bordes.
- **Operadores de Sobel:** Los operadores de Sobel son operadores diferenciales discretos utilizados para calcular los gradientes en una imagen. Estos operadores se aplican en las direcciones horizontal (x) y vertical (y) para obtener la magnitud y la dirección del cambio de intensidad en cada punto de la imagen. En el algoritmo de Canny, los operadores de Sobel se utilizan para calcular los gradientes x e y, que son fundamentales para la detección de bordes.
- El operador de Sobel en la dirección horizontal (x) se basa en un kernel que realiza una comparación entre el píxel central y sus vecinos en la misma fila. Este kernel tiene una configuración que resalta los cambios bruscos en la intensidad en sentido horizontal. De manera similar, el operador de Sobel en la dirección vertical (y) utiliza un kernel que realiza una comparación entre el píxel central y sus vecinos en la misma columna. Este kernel está diseñado para resaltar los cambios bruscos en la intensidad en sentido vertical.

El kernel del operador de Sobel en cada dirección es una combinación de dos kernels. Por un lado, se utiliza un kernel que realiza una comparación de los bits vecinos para resaltar los cambios bruscos en la intensidad. Por otro lado, se aplica un promedio ponderado de la comparación para obtener una respuesta más suave y precisa. Esta combinación de kernels permite detectar los cambios de intensidad en diferentes direcciones y calcular los gradientes x e y de manera eficiente.



$$Sx = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}; Sy = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

- **Supresión de no máximos:** La supresión de no máximos es una etapa importante en el algoritmo de Canny. Después de calcular los gradientes x e y, se realiza la supresión de no máximos para identificar los bordes potenciales en la imagen. Consiste en comparar la magnitud del gradiente en cada punto con la magnitud de los puntos vecinos a lo largo de la dirección del gradiente. Si la magnitud del gradiente en un punto es mayor que las magnitudes de los puntos vecinos en esa dirección, se considera un borde potencial. De esta manera, se resaltan los máximos locales en la dirección del gradiente y se suprimen los demás píxeles que no representan bordes.

La magnitud y ángulo del gradiente se pueden calcular con la siguiente ecuación:

$$G_{M_{i,j}} = \sqrt{G_{X_{i,j}}^2 + G_{Y_{i,j}}^2}$$

$$G_{A_{i,j}} = \text{atan2}(G_{Y_{i,j}}, G_{X_{i,j}})$$

El algoritmo de detección de bordes de Canny se basa en el uso de filtros Gaussiano, operadores de Sobel y la supresión de no máximos. El filtro Gaussiano suaviza la imagen para eliminar el ruido, los operadores de Sobel calculan los gradientes en las direcciones horizontal y vertical, y la supresión de no máximos identifica los bordes potenciales. Estos bordes potenciales serán utilizados posteriormente en el algoritmo para obtener los bordes finales mediante técnicas de umbral y rastreo de bordes por histéresis.

## Desarrollo

En el desarrollo de la práctica se implementó un algoritmo de detección de bordes utilizando el detector de bordes de Canny. A continuación, se describirá cada etapa del algoritmo y su función dentro del proceso de detección de bordes.

### 1. Preparación del entorno:

- Se importaron las bibliotecas necesarias, como cv2 (OpenCV) y numpy, para el procesamiento de imágenes y cálculos matemáticos.
- Se definió una función para realizar la convolución bidimensional (filtro Gaussiano) sobre una imagen.



2. Obtención del kernel Gaussiano:

- Se implementó una función para generar el kernel Gaussiano, que es utilizado para suavizar la imagen y reducir el ruido. El kernel se calcula en base a la dimensión (k) y la desviación estándar (sigma) especificados.

3. Cálculo de los gradientes:

- Se implementó una función para aplicar los operadores de Sobel en las direcciones horizontal (x) y vertical (y). Estos operadores permiten calcular los gradientes en cada punto de la imagen, proporcionando información sobre la dirección y la magnitud del cambio de intensidad de los píxeles.

4. Obtención de la magnitud y el ángulo:

- Se implementó una función para calcular la magnitud y el ángulo del gradiente en cada punto de la imagen, utilizando los gradientes obtenidos en el paso anterior.

5. Supresión de no máximo:

- Se implementó una función para aplicar la supresión de no máximo, que consiste en comparar la magnitud del gradiente en cada punto con sus vecinos en la dirección del ángulo del gradiente. Se seleccionan los puntos que representan máximos locales en la dirección del gradiente como bordes potenciales.

6. Umbral doble y detección de bordes:

- Se implementó una función para aplicar el umbral doble a los bordes potenciales, distinguiendo entre bordes fuertes y débiles. Los bordes con una magnitud del gradiente superior al umbral alto se consideran bordes fuertes, mientras que los bordes con una magnitud del gradiente entre el umbral bajo y el umbral alto se consideran bordes débiles.

7. Rastreo de bordes por histéresis:

- Se implementó una función para realizar el rastreo de bordes por histéresis, que consiste en conectar los bordes débiles a los bordes fuertes. Los bordes débiles que están conectados a bordes fuertes se consideran bordes finales, lo que permite completar los bordes y mantener la continuidad de las estructuras detectadas en la imagen.

8. Interfaz de usuario y ejecución del algoritmo:



- Se creó una interfaz de usuario mediante trackbars para ajustar los parámetros del algoritmo, como el tamaño del kernel, la desviación estándar, los umbrales de detección y otros parámetros.
- Se capturó el video en tiempo real desde la cámara y se aplicó el algoritmo de detección de bordes a cada fotograma.
- Se mostraron las imágenes resultantes de cada etapa del proceso de detección de bordes, incluyendo el fotograma original, la imagen filtrada, los gradientes en las En el desarrollo de la práctica se implementó un algoritmo de detección de bordes utilizando el detector de bordes de Canny. A continuación, se describirá cada etapa del algoritmo y su función dentro del proceso de detección de bordes.

9. Finalización del programa:

- El programa se detiene si se presiona la tecla "Esc" y se liberan los recursos de la cámara y las ventanas de visualización.

Este programa implementa el algoritmo de detección de bordes de Canny y proporciona una interfaz interactiva para ajustar los parámetros del algoritmo en tiempo real.

## Análisis de resultados

En cada una de las ventanas mostradas durante la ejecución del programa, se presentan diferentes etapas del proceso de detección de bordes. A continuación, se describen los resultados que se pueden observar en cada ventana:

1. Ventana "Original":

En esta ventana se muestra el fotograma original capturado desde la cámara en tiempo real. Es la imagen de entrada que se utiliza para la detección de bordes.



2. Ventana "Filtrada":

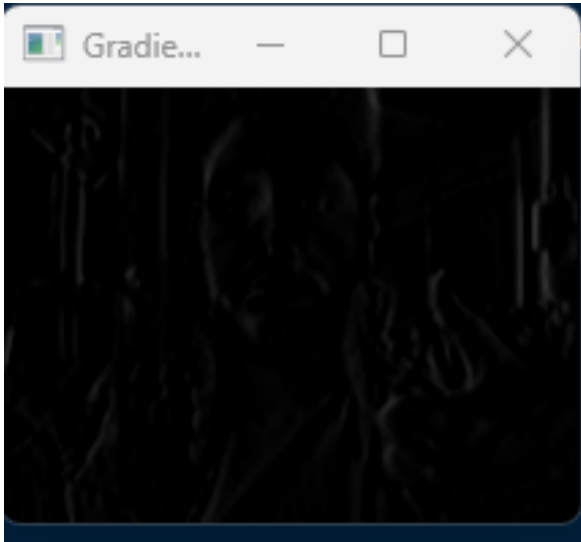
En esta ventana se muestra la imagen filtrada, obtenida aplicando el filtro Gaussiano sobre la imagen en escala de grises. La imagen filtrada suaviza la imagen original y reduce el ruido, lo cual es importante para obtener una detección más precisa de los bordes.



3. Ventana "Gradiente X":

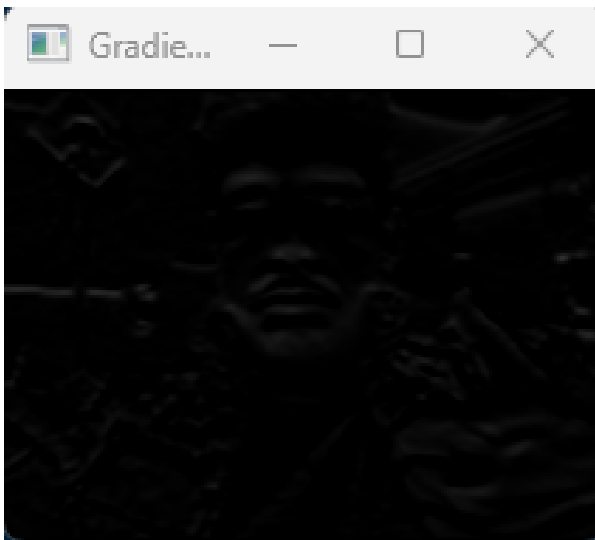
En esta ventana se visualizan los gradientes calculados en la dirección horizontal (x) utilizando el operador de Sobel. Los gradientes x representan los cambios de intensidad de los píxeles en sentido horizontal.





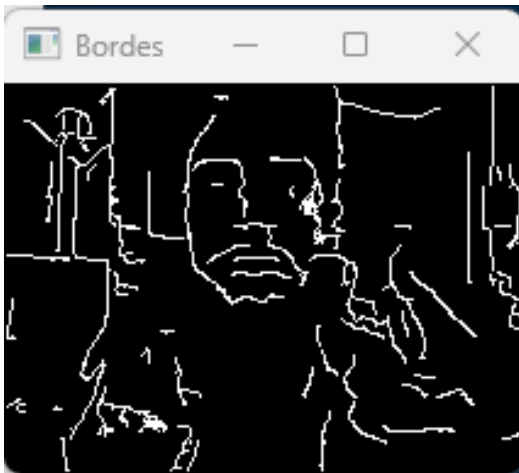
4. Ventana "Gradiente Y":

En esta ventana se muestran los gradientes calculados en la dirección vertical (y) utilizando el operador de Sobel. Los gradientes y representan los cambios de intensidad de los píxeles en sentido vertical.



5. Ventana "Bordes":

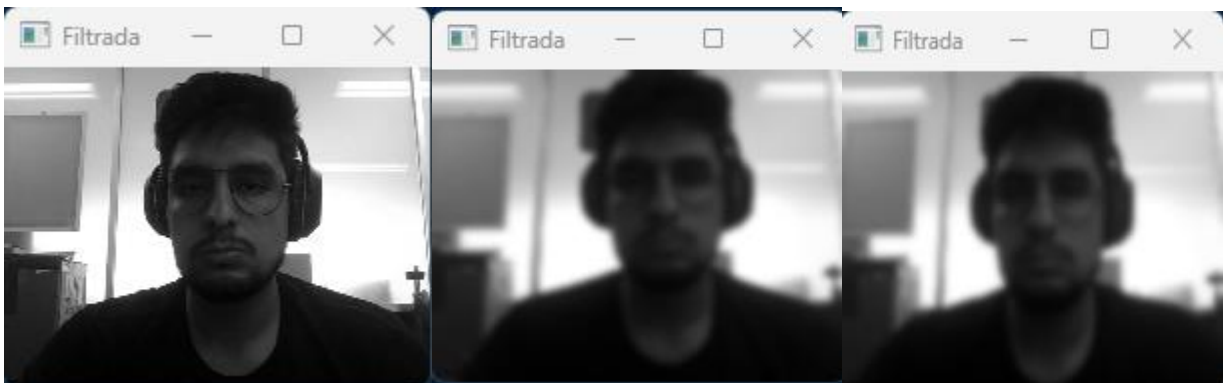
En esta ventana se presentan los bordes detectados después de aplicar el umbral doble y el rastreo de bordes por histéresis. Los bordes fuertes se muestran como píxeles blancos, mientras que los bordes débiles se representan en tonos de gris. Esta ventana muestra los bordes finales resaltados en la imagen.



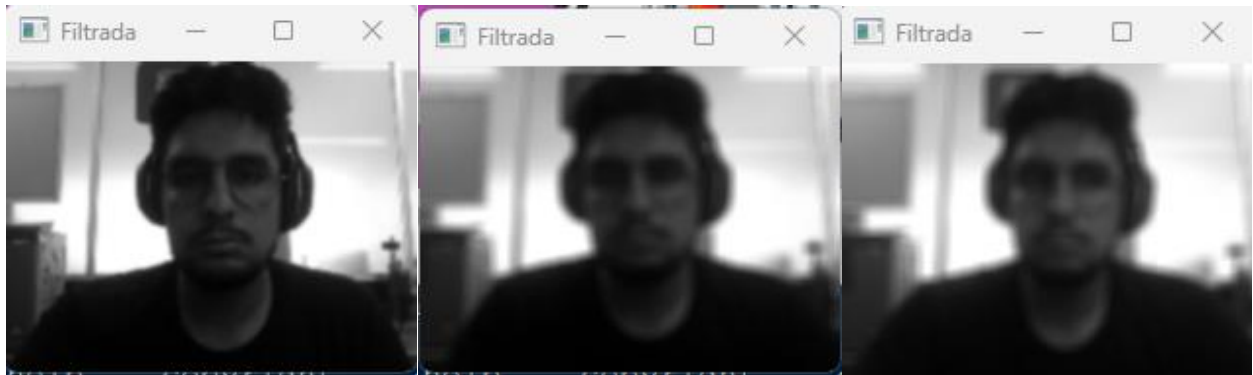
En el proceso de experimentación y observación durante la ejecución del algoritmo de detección de bordes de Canny, se realizaron diferentes ajustes a los parámetros clave, como el valor de  $k$ ,  $\sigma$  y los umbrales de detección. A continuación, se describen las principales observaciones obtenidas durante este proceso:

6. Al variar el valor de  $k$  en el filtro Gaussiano, se pudo notar que a medida que se incrementa, la imagen filtrada tiende a suavizarse más. Esto se traduce en una reducción en los detalles de la imagen y una mayor difuminación de los bordes presentes. Por otro lado, al disminuir el valor de  $k$ , se obtiene una imagen filtrada con una apariencia más nítida y con bordes más definidos.

A continuación, se muestran los resultados con  $k=1$ ,  $k=5$ ,  $k=10$ :

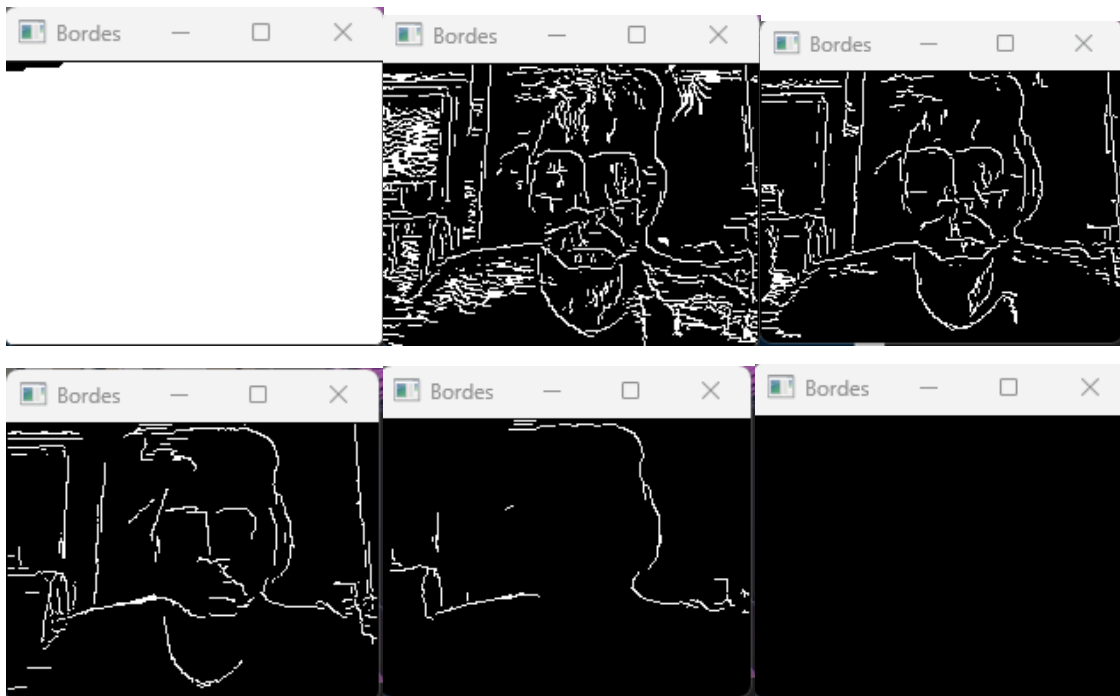


Mientras que al variar  $\sigma$  sucede algo parecido, manteniendo  $k=4$  y con  $\sigma=1$ ,  $\sigma=5$  y  $\sigma=10$

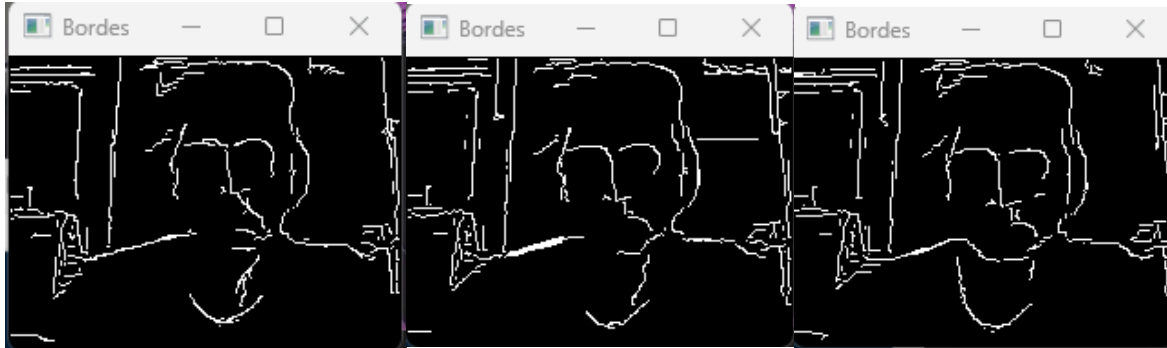


7. Al ajustar los umbrales de detección, se notaron cambios significativos en la cantidad y calidad de los bordes detectados. Al disminuir el umbral bajo, se permitió la detección de bordes con magnitudes de gradiente más bajas, lo que resultó en la aparición de bordes más sutiles y finos en la imagen. Por otro lado, al aumentar el umbral bajo, se estableció un límite más alto para considerar un píxel como borde, lo que resultó en una detección más estricta y la aparición de bordes más prominentes.

Primero se muestran los resultados del umbral alto en lo mas bajo y subiéndolo poco a poco:



Ahora hacemos lo mismo con el umbral bajo manteniendo el umbral alto en un nivel aceptable con bordes que muestren mucha información, pero no la suficiente para considerar pequeños detalles como bordes:



El ajuste del umbral alto demostró tener un impacto significativo en la detección de bordes. Al disminuir el umbral alto, se consideraron más píxeles como bordes, lo que resultó en una detección más sensible y una mayor cantidad de bordes detectados, incluyendo detalles más sutiles en la imagen. Por otro lado, al aumentar el umbral alto, la detección de bordes se volvió más restrictiva, considerando solo aquellos píxeles con magnitudes de gradiente muy altas como bordes. Esto llevó a una detección más precisa pero potencialmente omitiendo bordes más sutiles.

## Conclusiones

En esta práctica se implementó el algoritmo de detección de bordes de Canny utilizando técnicas de filtrado y operadores de gradiente. A través de la experimentación y observación de los resultados obtenidos, se pudieron destacar varias conclusiones importantes.

El algoritmo de Canny demostró ser una herramienta eficaz para la detección precisa y robusta de bordes en imágenes. Su capacidad para suprimir el ruido, detectar bordes finos y mantener una respuesta precisa en presencia de discontinuidades en la imagen lo convierte en un enfoque ampliamente utilizado en el procesamiento de imágenes.

La utilización de operadores de Sobel permitió calcular los gradientes en la imagen, proporcionando información valiosa sobre la dirección y magnitud del cambio de intensidad de los píxeles. Estos gradientes fueron utilizados para resaltar los bordes en la etapa de supresión de no máximo, donde se compararon las magnitudes del gradiente con sus vecinos para determinar los bordes potenciales.



La aplicación del umbral doble permitió distinguir entre bordes fuertes y débiles, eliminando el ruido y resaltando los bordes más significativos en la imagen. El proceso de rastreo de bordes por histéresis aseguró la conexión de los bordes débiles a los bordes fuertes, manteniendo la continuidad de las estructuras detectadas.

Durante la experimentación, se observó que los ajustes de los parámetros, como el tamaño del kernel en el filtro Gaussiano y los umbrales de detección, tuvieron un impacto significativo en los resultados. La selección adecuada de estos parámetros permitió adaptar el algoritmo a diferentes escenarios y objetivos de detección de bordes, obteniendo resultados óptimos.

## Bibliografía

1. Canny, J. (1986). A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence.