

1. Características de la imagen

La imagen seleccionada para el ejercicio se muestra en la figura 1



Figura 1: Imagen seleccionada

1.1. Descripción informativa de la escena

Se observa el interior de un vagón del metro en movimiento. La imagen está iluminada con luz natural que entra por las ventanas y está enfocada sobre uno de los tubos pasamanos verticales.

1.2. Descripción técnica

En las tablas 1 y 2 se muestran las características de la imagen y de la cámara, respectivamente.

Cuadro 1: Características de la imagen

Característica	Dato
Exposición	1/60 seg a f/4.5
Distancia focal	28 mm
Rango ISO	ISO 125
Flash	Sin flash
Dimensión	6000 x 4000 px
Formato	tif

Cuadro 2: Características de la cámara

Característica	Dato
Marca	Sony
Modelo	ILCE-6000
Lente	E PZ 16-50mm F3.5-5.6 OSS

Finalmente, en la figura 2 se observa su histograma

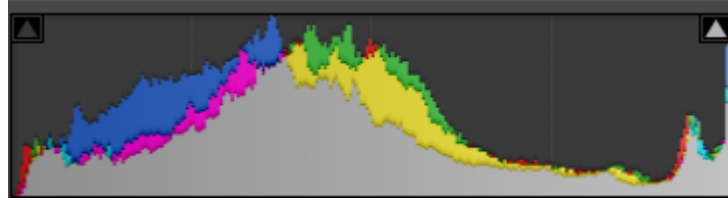


Figura 2: Histograma de imagen

2. Algoritmo de detección de Bordes

Se realizará la detección de bordes en matlab y por medio de los métodos de Canny y Sobel. Se lee la imagen de entrada con la función 'imread', después se pasa a a escala de grises con la función 'rgb2gray'. Finalmente se utiliza la función edge, se especifica, el método y se provee un valor de umbral entre 0.1 y 0.9. Se muestra a continuación

```

1 close all;
2 clc;
3 input = imread('metro.tif');
4
5 umbral_c = 0.12;
6
7 gray = rgb2gray(input);
8 borde_canny = edge(gray, 'canny', umbral_c);
9 borde_sobel = edge(gray, 'sobel');
10
11 figure;
12 imshow(borde_canny)
13 title('M todo Canny. Detecci n de bordes.');
```

```

14
15 figure;
16 imshow(borde_sobel)
17 title('M todo Sobel. Detecci n de bordes ');
```

Listing 1: Algoritmos implementados en Matlab

3. Resultados

En la figura 3 se observa la imagen en escala de grises



Figura 3: Conversión a escala de grises

3.1. Método Sobel

En la figura 4 se observa el resultado del algoritmo de detección de Sobel, sin especificar un umbral.

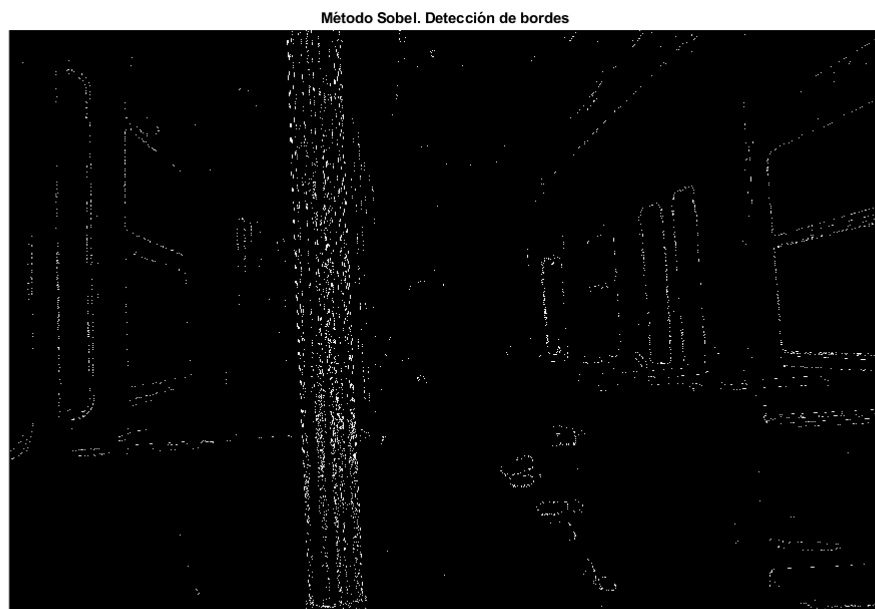


Figura 4: Resultado de método Sobel sin especificar umbral

3.2. Método Canny

En la figura 5 se observa el resultado del algoritmo de detección de Canny, sin especificar un umbral.

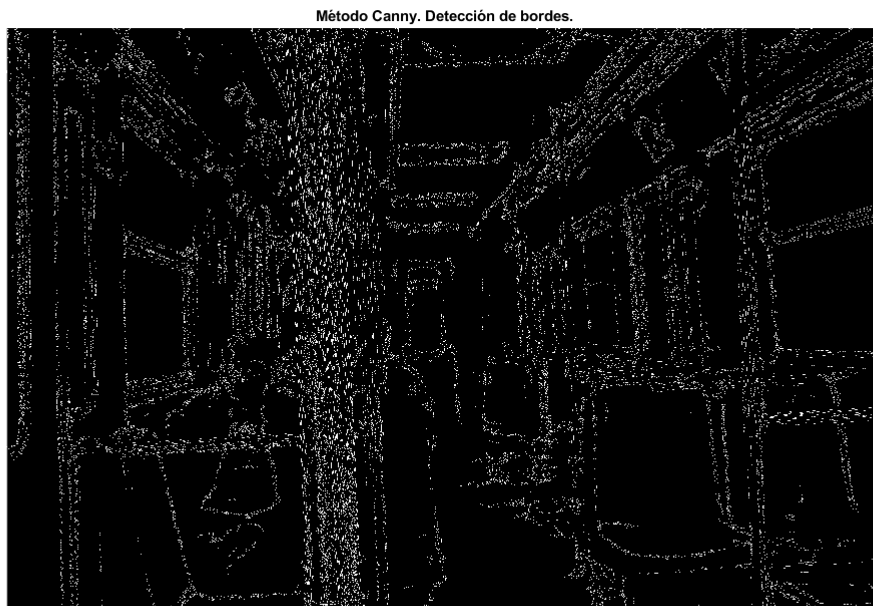


Figura 5: Resultado de método Canny sin especificar umbral

Buscando disminuir la cantidad de ruido se buscó un umbral para limitar la detección. El umbral seleccionado fue de 0.12 y el resultado se muestra en 6

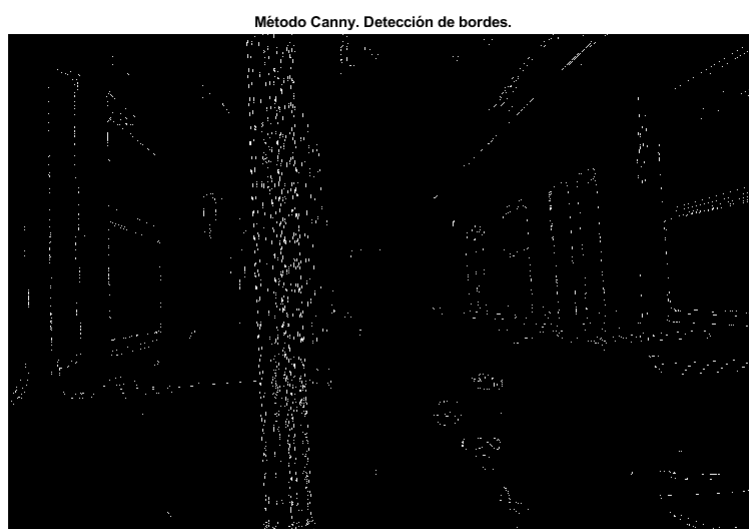


Figura 6: Resultado de método Canny con Umbral: 0.12

Después de obtener estos resultados se realizó una prueba con la misma imagen pero en resolución 480 x 320 px, en formato JPG. Adjunto este resultado en la figura 7

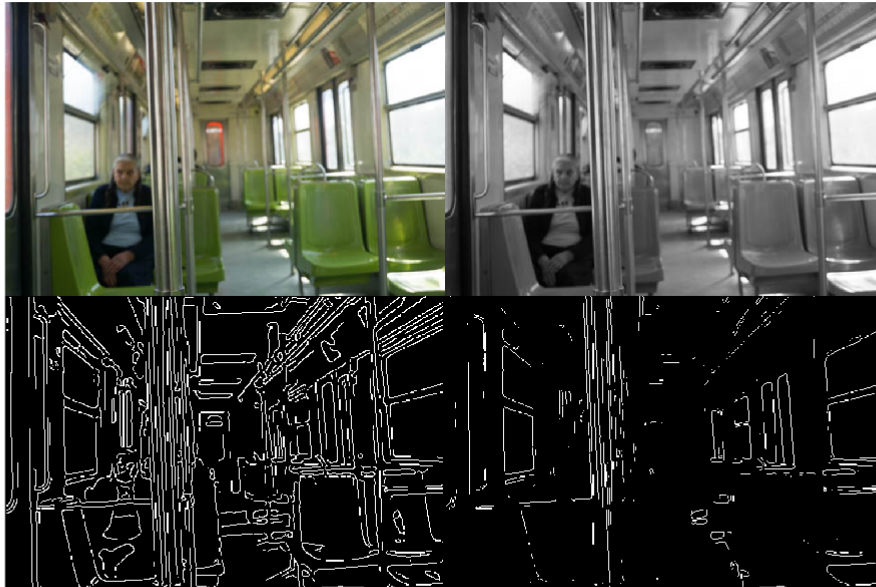


Figura 7: Resultado imagen de baja resolución. Abajo a la izquierda: Método Canny. Abajo a la derecha: Método Sobel.

4. Conclusiones

- El método de Sobel mostró un aceptable desempeño siendo capaz de detectar las ventanas y el tubo de acero que está en primer plano, por otro lado no fue capaz de detectar bordes ni en los asientos, ni de la figura humana que se observa.
- La cantidad de ruido en este algoritmo no me parece tan considerable
- El método de Canny detecto una gran cantidad de bordes y detalles, sin embargo, agrego algo de ruido.
- Los detalles de ruido en ambos algoritmos parecen ser causados por los fuertes reflejos de luz que crean areas sobreexpuestas en la imagen.
- El procesamiento de la imagen de baja resolución parece haber creado lineas más marcadas que en la de alta resolución, no estoy seguro si esto es causado por el reescalamiento que hace Matlab al desplegar la imagen de alta resolución.