

Práctica 4. Histograma

Reyes Hernández María Lucero

Romero Robles Juan Carlos

Introducción

Para el procesamiento de imágenes, se utiliza un concepto esencial que es el del histograma BGR cuya labor principal es mostrar de manera gráfica la distribución de cada tonalidad de los colores principales (Blue-Green-Red).

Por otro lado, existen transformaciones matemáticas que permiten visualizar más a fondo los bordes en las imágenes, entre ellas están las Derivadas de Gradiente y Sobel, la Transformada de Laplace y Canny. Además también con la Transformada Discreta de Fourier se puede descomponer la imagen de tal manera que se muestran altas o bajas frecuencias en la imagen.

En esta práctica, se añadió al proyecto de las prácticas anteriores las funciones de Gradiente y Sobel, las transformadas de Laplace y Canny y la derivada discreta de Fourier, las cuales están implementadas por la librería OpenCV, además se integró la posibilidad de poder visualizar el histograma correspondiente a una imagen entrada.

Antecedentes

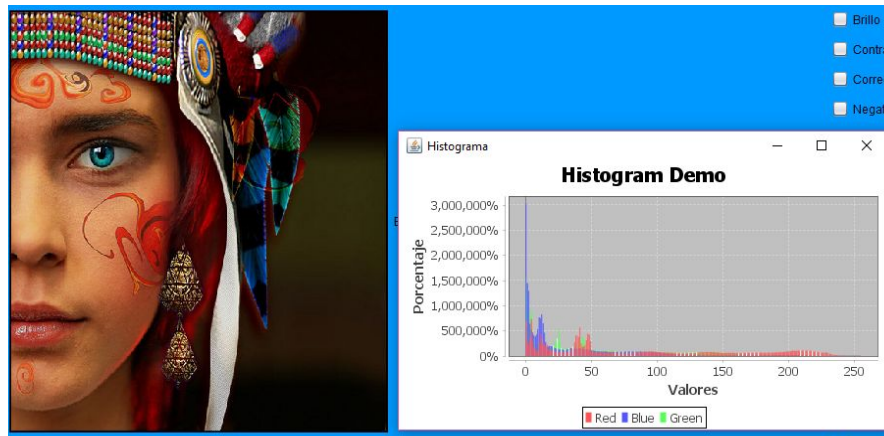
Por parte de la imagen digital se obtiene una matriz numérica que frecuentemente se representa de forma binaria es decir entre 1 y 0.

Los píxeles son puntos de color en una escala de grises, siendo de una gama de color monocromática. el píxel es la menor unidad homogénea en color que es parte de la imagen digital.

En las imágenes el mapa de bits, cada píxel se codifica mediante un conjunto de bits de longitud determinada (la profundidad de color); por ejemplo, puede codificarse un píxel con un byte (8 bits), de manera que cada píxel admite hasta 256 variaciones de color, de 0 a 255. En las imágenes llamadas de color verdadero, normalmente se usan tres bytes (24 bits) para definir el color de un píxel; es decir, en total se pueden representar unos 224 colores, esto es 16.777.216 variaciones de color.



En el procesamiento digital de imágenes, el **histograma** hace referencia a la frecuencia de blancos y negros, escalas de grises o colores RGB de la imagen. En el eje Y se representa el número de píxeles de la imagen, la cual en este caso tiene un total de 2100; en el eje X se representa la escala de colores, la cual se encuentra en el rango [0,255]. Además se aprecia la frecuencia de cada escala de color RGB por píxeles de la imagen.



Con el histograma se obtiene la frecuencia de colores RGB de la imagen por número de píxeles.

Por otra parte las **transformaciones de las imágenes**.

Las transformaciones de imágenes, algunos de estos métodos son artefactos que se utilizan para encontrar bordes de imagen y transformaciones que nos ayudan a encontrar líneas y círculos en una imagen.

Las transformaciones utilizadas para esta práctica son las siguientes:

- Derivadas de Gradiente y Sobel
- Transformada de Laplace y Canny
- Transformada Discreta de Fourier.

Derivadas de Gradiente y Sobel

El operador Sobel es utilizado en procesamiento de imágenes, especialmente en algoritmos de detección de bordes.

El operador Sobel calcula el gradiente de la intensidad de una imagen en cada punto (píxel). Así, para cada punto, este operador da la magnitud del mayor cambio posible, la dirección de éste y el sentido desde oscuro a claro. El resultado muestra cómo de abruptamente o suavemente cambia una imagen en cada punto analizado y, en consecuencia, cuán probable es que éste represente un borde en la imagen y, también, la orientación a la que tiende ese borde.

Matemáticamente, el gradiente de una función de dos variables para cada punto es un vector bidimensional cuyos componentes están dados por las primeras derivadas de las direcciones verticales y horizontales. Para cada punto de la imagen, el vector gradiente apunta en dirección del incremento máximo posible de la intensidad, y la

magnitud del vector gradiente corresponde a la cantidad de cambio de la intensidad en esa dirección.

Utilizando las funciones de openCV Sobel ya está declarado de la siguiente manera:

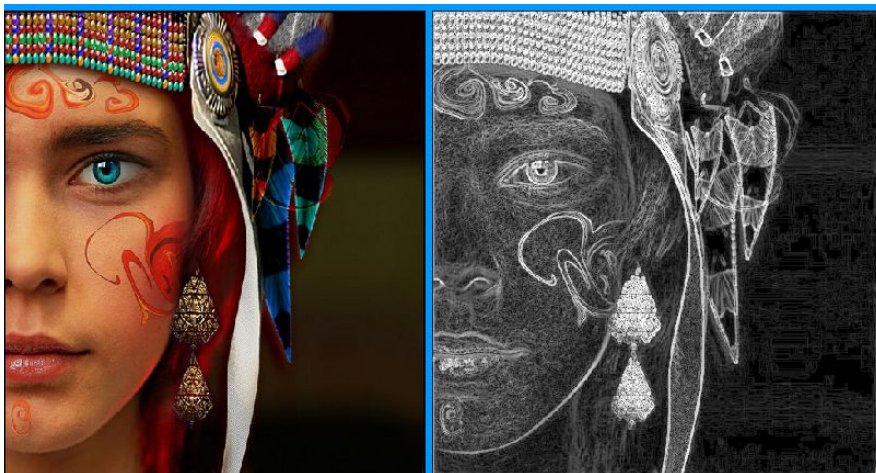
```
void cv::Sobel (InputArray src ,  
               OutputArray dst ,  
               int ddepth ,  
               int dx ,  
               int dy )
```

Parámetros

src	imagen de entrada.
dst	Imagen de salida
ddepth	Profundidad de imagen de salida
dx	orden de la derivada x.
dy	orden de la derivada y.

Los operadores de Sobel con el resultado es más o menos resistente al ruido. La mayoría de las veces, la función se llama con (xorder = 1, yorder = 0, ksize = 3) o (xorder = 0, yorder = 1, ksize = 3) para calcular la primera derivada de la imagen x o y. El primer caso corresponde a un kernel de:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



Transformada de Laplace y Canny

Transformada de Laplace

Los filtros laplacianos no existen para la primera derivada. El Laplaciano es un operador de tipo escalar formado por las segundas derivadas de ambas direcciones (X e Y). Por lo tanto es capaz de detectar bordes, pero no en máximos y mínimos (como el caso de los filtros Sobel) sino en los cruces por cero. Este filtro se desarrolló por completo empíricamente y posteriormente se le encontró parecido con la segunda derivada de la Gaussiana.

```
void cv::laplaciano (    InputArray  src ,
                        OutputArray dst ,
                        int          ddepth )
```

Parámetros

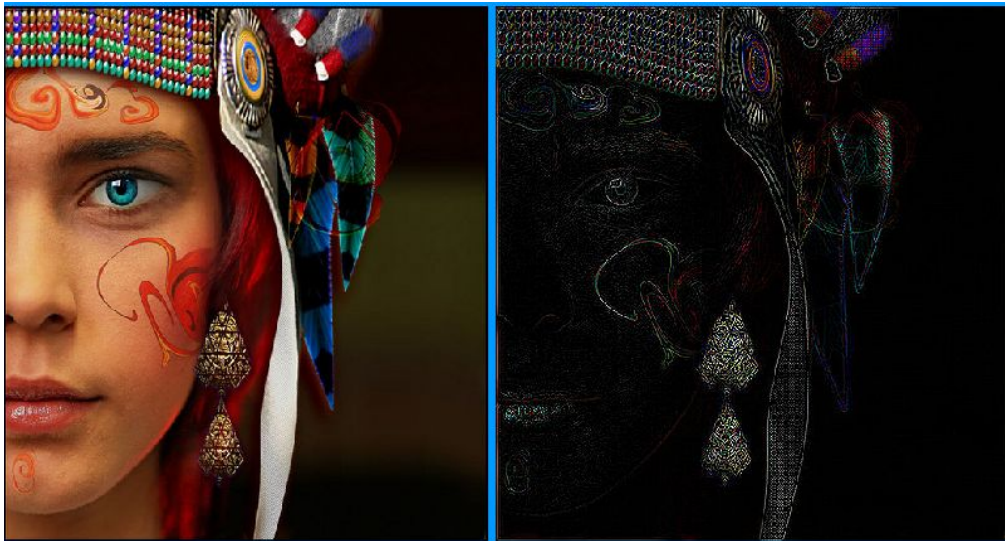
src Fuente de imagen.

dst Imagen de destino.

ddepth Profundidad deseada de la imagen de destino.

Esto se hace cuando ksize > 1. Cuando ksize == 1, el laplaciano se calcula filtrando la imagen con la siguiente apertura 3 × 3

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



Transformada de Canny

Utiliza un algoritmo de múltiples etapas para detectar una amplia gama de bordes en imágenes. Lo más importante es que Canny también desarrolló una teoría computacional acerca de la detección de bordes que explica por qué la técnica funciona.

El propósito de Canny era descubrir el algoritmo óptimo de detección de bordes. Para que un detector de bordes pueda ser considerado óptimo debe cumplir los siguientes puntos:

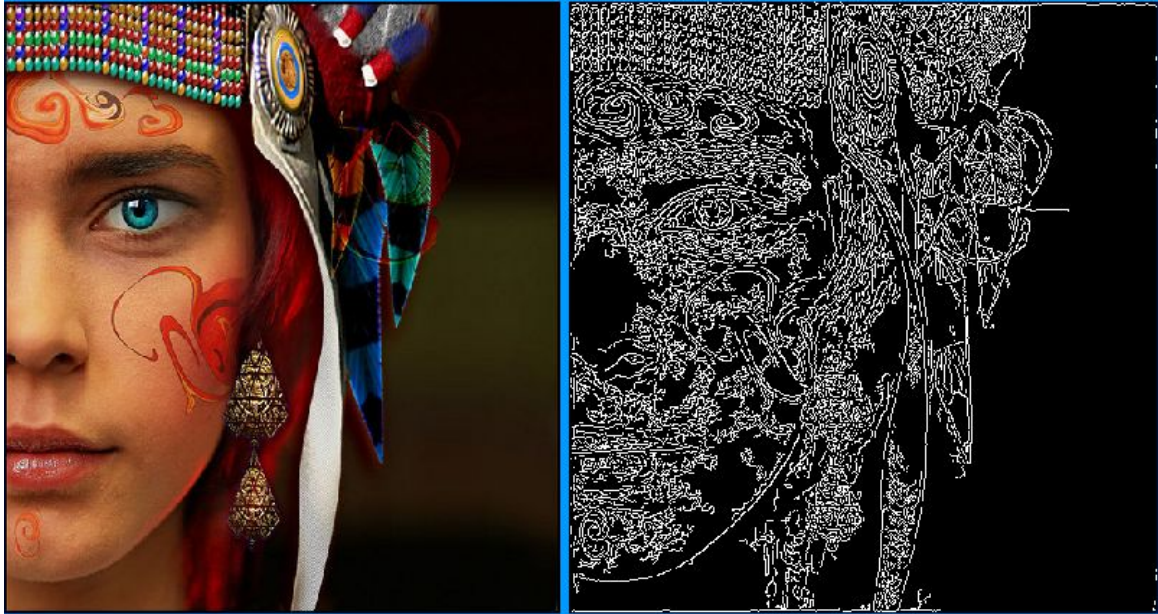
- Buena detección- el algoritmo debe marcar el mayor número real en los bordes de la imagen como sea posible.
- Buena localización- los bordes de marca deben estar lo más cerca posible del borde de la imagen real.
- Respuesta mínima - El borde de una imagen sólo debe ser marcado una vez, y siempre que sea posible, el ruido de la imagen no debe crear falsos bordes.

```
void cv::Canny (InputArray imagen ,
                OutputArray bordes ,
                doble umbral1 ,
                doble umbral2 ,
                intt apertureSize = 3,
                bool L2gradient =false
```

Parámetros

imagen	Imagen de entrada
bordes	mapa de borde de salida;
umbral1	Primer umbral para el procedimiento de histéresis.
umbral2	Segundo umbral para el procedimiento de histéresis.
Tamaño de apertura	Tamaño de apertura para el operador Sobel.

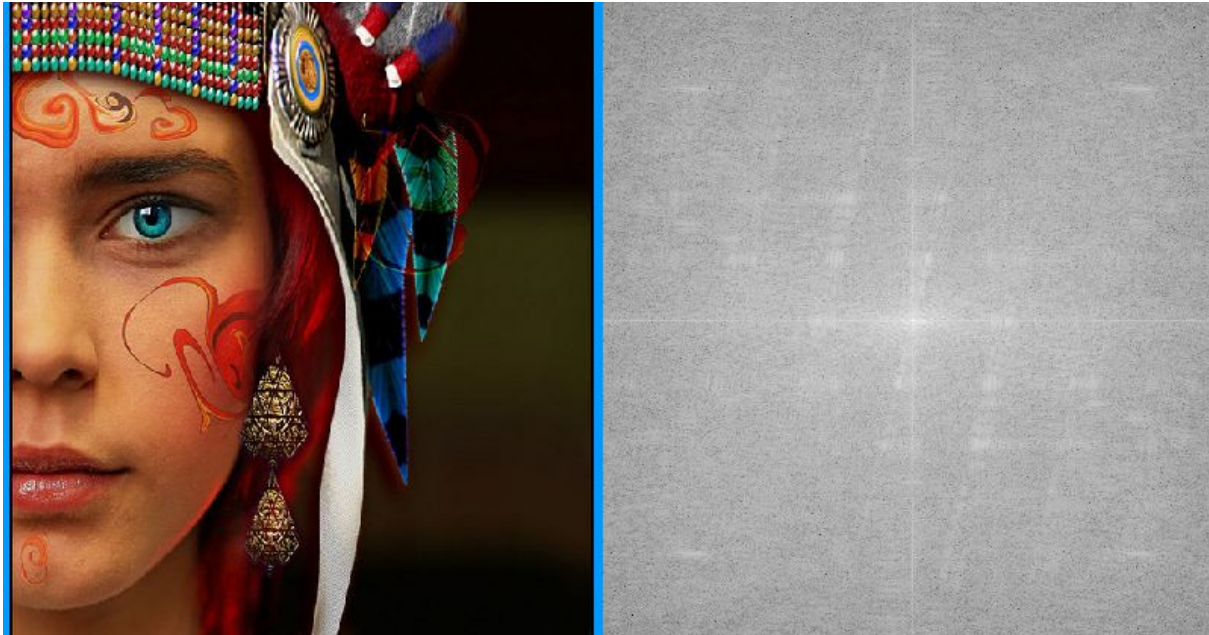
L2gradient una bandera, (L2gradient = false) que indica si se debe usar una norma más precisa para calcular la magnitud del gradiente de la imagen (L2gradient = true), o si el valor predeterminado es suficiente



Transformada Discreta de Fourier.

La transformada de Fourier descompondrá una imagen en sus componentes sinusoidal y coseno. En otras palabras, transformará una imagen de su dominio espacial a su dominio de frecuencia. La idea es que cualquier función puede aproximarse exactamente con la suma de las funciones infinitas de los senos y los cosenos.

Aquí f es el valor de la imagen en su dominio espacial y F en su dominio de frecuencia. El resultado de la transformación son números complejos. Mostrar esto es posible ya sea a través de una imagen real y una imagen compleja o bien a través de una imagen de magnitud y fase. Sin embargo, a lo largo de los algoritmos de procesamiento de imágenes, solo la imagen de magnitud es interesante, ya que contiene toda la información que necesitamos sobre la estructura geométrica de las imágenes. Sin embargo, si pretende realizar algunas modificaciones de la imagen en estas formas y luego necesita volver a transformarla, deberá conservar ambas.



Conclusiones

Con cada uno de los procesos que se realizaron para la práctica junto con los anteriores conocimos un poco más sobre las estructuras que hay en una imagen y el cómo se pueden manipular la imagen original hasta llegar a distintos resultados de dicha imagen desde aplicar un filtro, o rotar, o hacer una transformación a las imágenes que como vemos en cada transformación que se hizo el cómo se va perdiendo la forma de la imagen original.

En esta práctica en específico, utilizamos filtros que nos ayudan a detectar bordes dentro de la imagen, notamos que el algoritmo de detección de bordes que arroja mejores resultados es el de la derivada de Sobel, ya que resalta los bordes más esenciales de la imagen.