

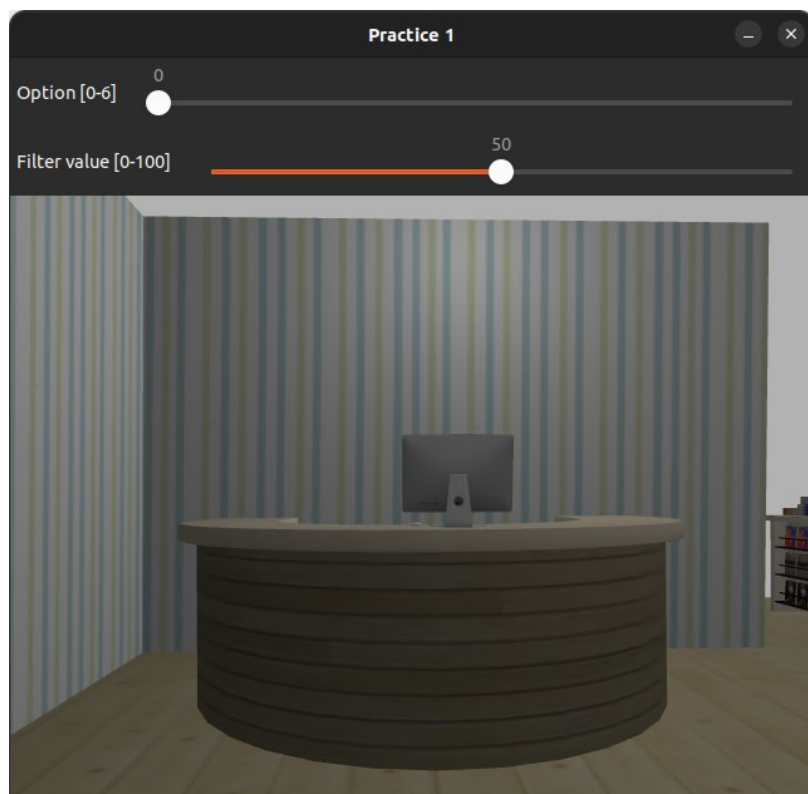
Práctica 1 – Espacios de color y DFT

Este ejercicio tiene como objetivo que familiarizar con OpenCV, el trabajo a bajo nivel de los píxeles de una imagen, su librería para cambios de espacios de color y sus transformaciones del dominio y espacial. Todo ello visto en el **Tema 2: Formación de la imagen** y **Tema 3: Transformación del dominio y espacial**.

Puntos totales posibles del ejercicio: 10

Instrucciones

Utilizando el simulador con Tiago, se pide crear un programa que trabaje con la imagen y muestre en la parte superior varios controles deslizantes (sliders) como los de la figura:



Para ello, utilizaremos las siguientes funciones:

```
// create Trackbar and add to a window
cv::createTrackbar("trackbar_text", "window_name", nullptr, max_value, 0);
// set Trackbar's value
cv::setTrackbarPos("trackbar_text", "window_name", value);
// get Trackbar's value
cv::getTrackbarPos("trackbar_text", "window_name");
```

donde **"trackbar text"** es el texto que queremos que aparezca en el slider; **"window name"** es el nombre de la ventana a la que se añade el slider; **max_value** es el valor máximo que puede alcanzar el slider; y **value** es el valor que se le asigna al slider (normalmente es el valor que tendrá inicialmente). Hay que tener cuidado de que los nombres sean los mismos en las distintas funciones.

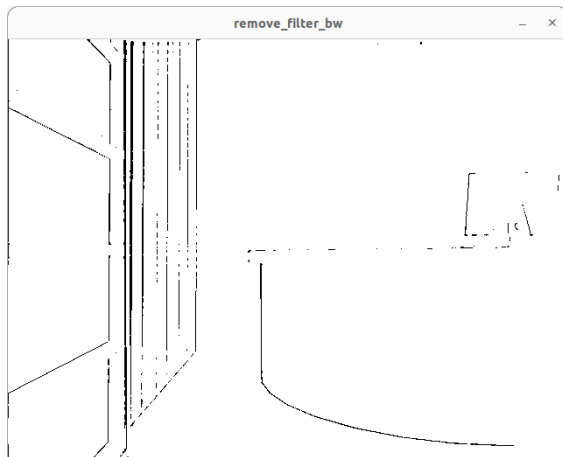
Estos sliders se tienen que añadir una única vez. Para ello es posible crear la ventana inicialmente con la siguiente función:

```
// create a window  
cv::namedWindow("window_name");
```

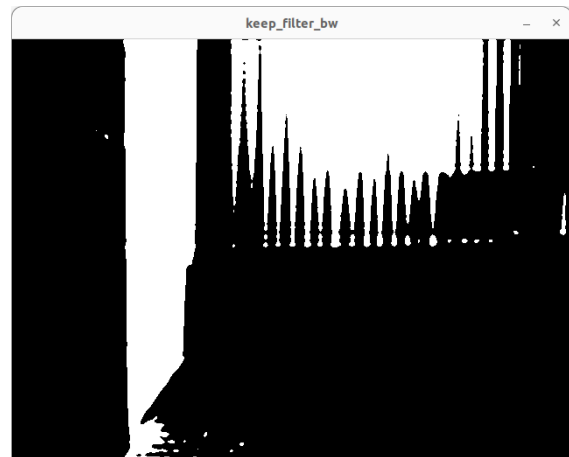
Se pide que en cada una de las 7 opciones se haga lo siguiente con la imagen **BGR** del Tiago:

- Opción 0: Mostrar la imagen en formato de color RGB.
- Opción 1: Mostrar la imagen en formato de color HSI haciendo la conversión píxel a píxel.
- Opción 2: Mostrar la imagen resultante de generar la imagen HSV utilizando la función **cvtColor** de OpenCV, y restando la imagen HSI calculada en la opción 1. Esta diferencia se realizará a nivel de canales, es decir, la imagen resultado [ABC] será la consecuencia de realizar [HSV-HSI], siendo [A=H-H], [B=S-S], [C=V-I].
- Opción 3: Mostrar la imagen del espectro de la **Transformada de Fourier Discreta**.
- Opción 4: Aplicar un filtro al espectro de Fourier que **mantenga** únicamente las frecuencias horizontales predominantes, y mostrar la imagen resultado de hacer la transformada inversa.
- Opción 5: Aplicar un filtro al espectro de Fourier que **elimine** únicamente las frecuencias horizontales predominantes, y mostrar la imagen resultado de hacer la transformada inversa.
- Opción 6: Aplicar una transformación píxel a píxel con un operador de umbral y realizar la **operación lógica OR**. Para ello, se realizará una transformación de umbral de la imagen filtrada en la **opción 4**, y suponiendo que la imagen está normalizada entre 0 y 1, se le aplicará el criterio de **umbralización de 0.6**, de manera que todos los valores superiores a este umbral sean 255 y el resto 0. Para la segunda imagen, partimos de la imagen filtrada en la **opción 5** y se le aplica el criterio anterior con un **umbral de 0.4**. Una vez obtenidas las dos imágenes, se realizará la transformación lógica OR utilizando OpenCV.

Esta última opción viene de aplicar la operación lógica OR sobre las siguientes imágenes, obtenidas de aplicar una transformación de píxel con un operador umbral. Supongamos que se obtienen las siguientes imágenes:



Mantener frecuencias y umbral $p = 0.4$



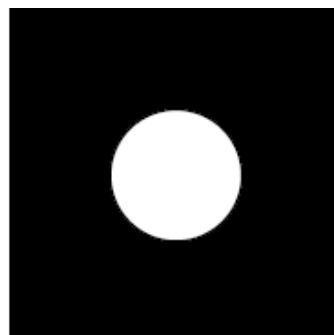
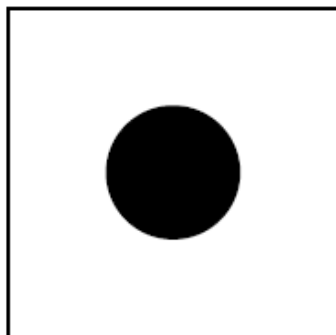
Eliminar frecuencias y umbral $p = 0.6$

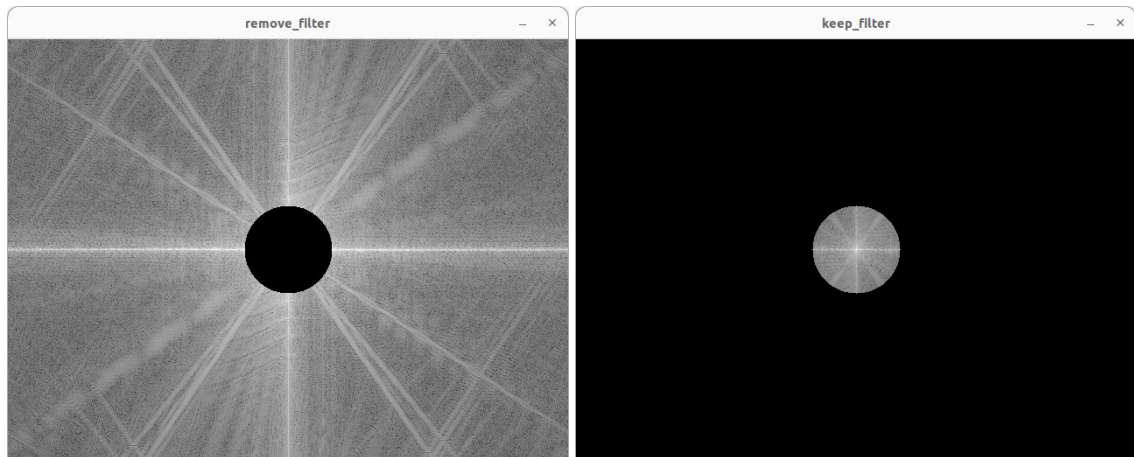
Se facilita la siguiente función, la cual se encarga de una vez obtenida la imagen con el filtro que se desea aplicar (mantener o eliminar), multiplica dicha imagen de filtro por la imagen del espectro de Fourier:

```
// Multiply fourier spectrum and filter  
mulSpectrums(fourier, filter, fourier, 0); // multiply 2 spectrums
```

La imagen de filtro puede generarse de muchas formas distintas. Una opción, es generar una imagen de igual tamaño que el espectro, la cuál será blanca o negra dependiendo del filtro a aplicar, y dibujando el patrón de las frecuencias de valor opuesto al fondo.

El **patrón que se dibuje** tendrá un **tamaño** que será variable de **0 a 100** para **ambos filtros**, y se modificará el slider correspondiente que **decrementará o incrementará el valor en 1** respectivamente. Suponiendo que nuestro patrón es un círculo, el resultado obtenido debe tener el siguiente aspecto antes de aplicar la multiplicación de los espectros:





Mediante el slider se incrementará/decrementará el radio, mientras que si es otra figura se modificará el alto/ancho/ambos (es diseño vuestro). Estas cuatro ventanas (las dos de los filtros y sus umbralizaciones) se mostrarán en caso de pulsar la **tecla d**, y se ocultarán al pulsarla de nuevo siempre que se esté visualizando la opción 6.

Para leer la tecla que se ha pulsado sobre la imagen, se debe usar una de las siguientes funciones:

```
// Read key
int key = cv::waitKey(int delay); // waiting time for a key pressed
int key = cv::pollKey();
```

Se recomienda utilizar el mundo **aws_bookstore** ya que contiene patrones que fácilmente son detectables.

Ayuda

waitKey: [enlace](#)

pollKey: [enlace](#)

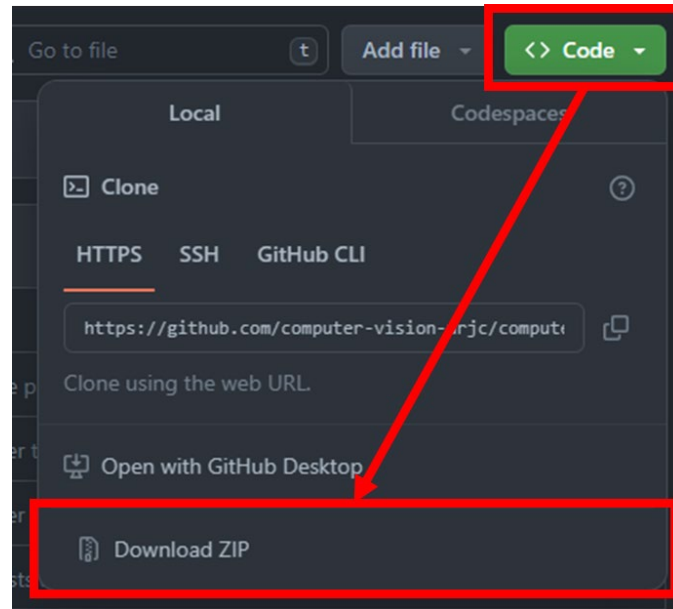
Memoria

Deberá modificarse el README y añadir un apartado para responder los siguientes puntos:

1. Adjunta una captura de la opción 2, e indica el por qué se ven esos colores en la imagen resultado de la resta entre HSV y HSI.
2. ¿Qué se observa en la imagen resultado de la opción 6 y en qué influye que se varíe el valor del filtro?

Entrega

La **entrega** consistirá en subir al **Aula virtual** el **archivo .zip** generado a través del repositorio de GitHub Classroom.



Para ello, será necesario que cada integrante del grupo entre en el siguiente [enlace](#), y se asocie con el usuario creado a partir de su email de la URJC. Una vez hecho esto, deberá entrar en el grupo de la asignatura, si no existe, uno de los miembros tendrá que crearlo, siendo el nombre del grupo el mismo que figura en el Aula Virtual.

La plantilla con el nodo ROS 2 proporcionada, deberá ser modificada para que el **nombre del paquete** sea **practica1-grupoX**, donde X es el número de grupo del Aula Virtual. Es importante **no modificar el archivo cabecera (.hpp)** de la plantilla que se proporciona, todo el **código** necesario deberá estar **incluido** únicamente en su **archivo fuente (.cpp)**.

Si no se cumplen los criterios anteriores, o se entrega un paquete que no compila, la calificación será 0.