Práctica 1 – Espacios de color y DFT

Este ejercicio tiene como objetivo familiarizarse 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

Cada integrante del grupo debe acceder al siguiente enlace y asociarse con el usuario creado a partir de su correo electrónico de la **URJC**.

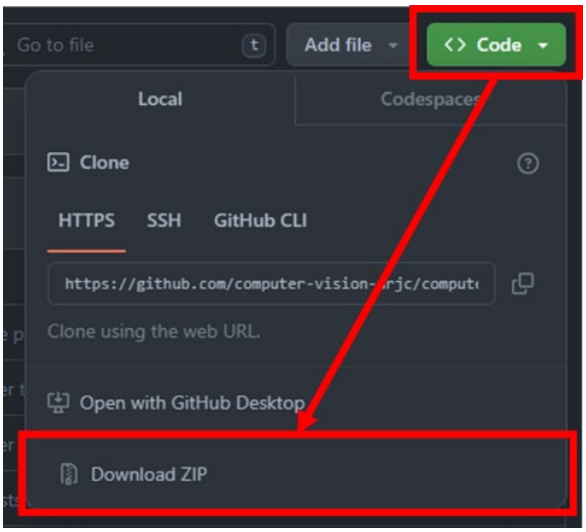
Una vez hecho esto, deberá unirse al grupo de la asignatura. Si dicho grupo no existe, uno de los miembros deberá crearlo, utilizando el **mismo nombre** que figura en el **Aula Virtual**.

📌 La plantilla con el nodo **ROS 2** proporcionada deberá modificarse para que el nombre del paquete sea: **practica1-grupoX**, donde **X** es el número de grupo asignado en el **Aula Virtual**.

⚠️ **Importante:** No se debe modificar el archivo de cabecera (.hpp) de la plantilla proporcionada. Todo el código necesario deberá implementarse únicamente en el archivo fuente (.cpp).

Entrega

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



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

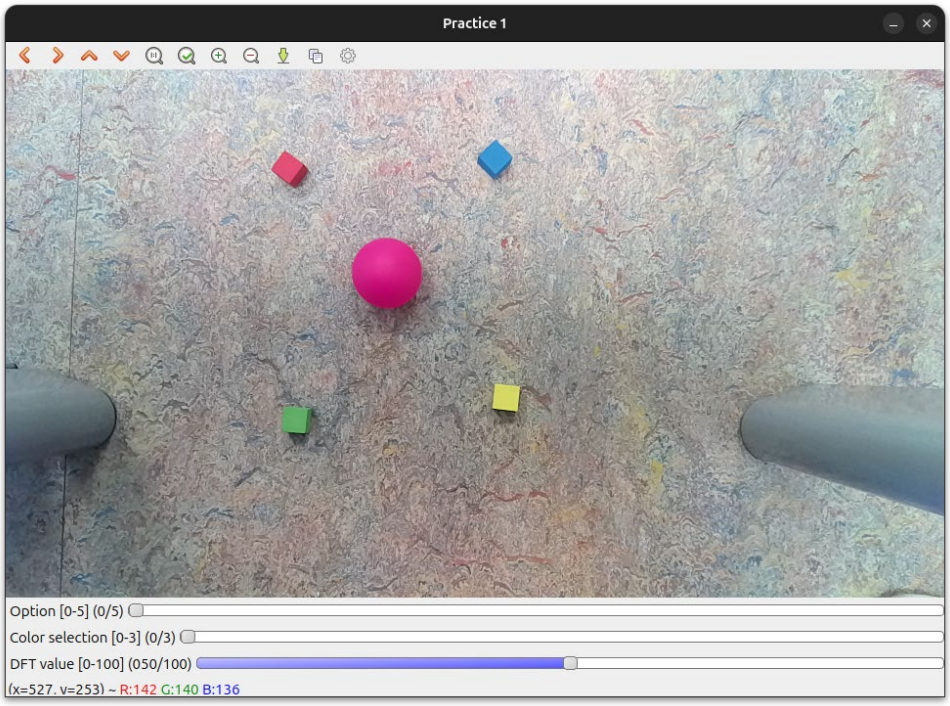
Enunciado

Para lanzar la cámara, se utilizará el siguiente comando a fin de mejorar el rendimiento:

```
// Camera launcher
ros2 launch oak_d_camera rgbd_stereo.launch.py only_rgb:=true use_rviz:=false
```

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:

1. **Option**: que irá de 0 a 5.
2. **Color selection**: que irá de 0 a 3.
3. **DFT value**: que irá de 0 a 100.



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");
```

Para cada una de las **5 opciones**, se debe realizar lo siguiente con la imagen **BGR**:

- **Opción 0:** Mostrar la imagen en formato de color **BGR**.
- **Opción 1:** Convertir la imagen al espacio de color **HSV** utilizando la función **cvtColor** y mostrar el resultado.
- **Opción 2:** Mostrar solo la **pelota rosa** aplicando un **filtro de color** sobre el **espacio HSV**.
- **Opción 3:** Dependiendo del valor del segundo *slider*, se mostrará únicamente uno de los **cubos** (0 – rojo, 1 – verde, 2 – azul, 3 – amarillo). Para ello, se aplicarán los **filtros de color** correspondientes en el **espacio BGR**.
- **Opción 4:** Aplicar un **filtro paso bajo** al espectro de Fourier y mostrar la imagen obtenida tras la transformada inversa. El tamaño del filtro dependerá del valor seleccionado en el tercer *slider*.
- **Opción 5:** Aplicar un **filtro paso alto** al espectro de Fourier y mostrar la imagen obtenida tras la transformada inversa. El tamaño del filtro dependerá del valor seleccionado en el tercer *slider*.

En las opciones 4 y 5, 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 a filtrar 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 mediante el *slider* correspondiente que **decrementará o incrementará el valor en 1**.

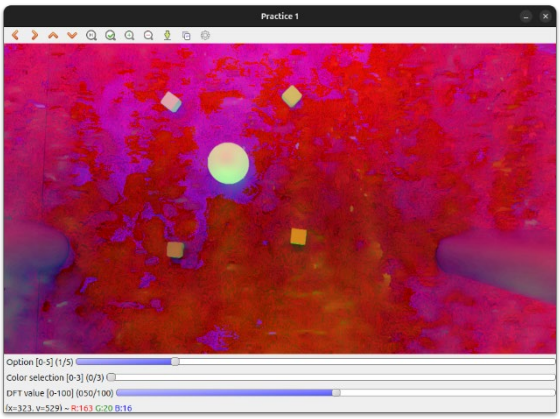
Se proporciona la siguiente función, que, una vez creado el filtro deseado (**paso alto** o **paso bajo**), multiplica la imagen del filtro por el espectro de Fourier de la imagen original:

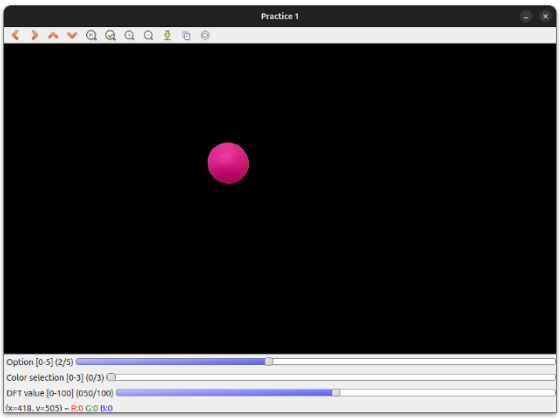
```
// Multiply fourier spectrum and filter
mulSpectrums(fourier, filter, fourier, 0); // multiply 2 spectrums
```

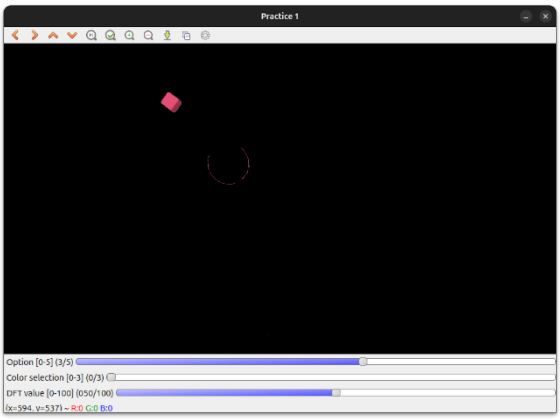
Ayuda

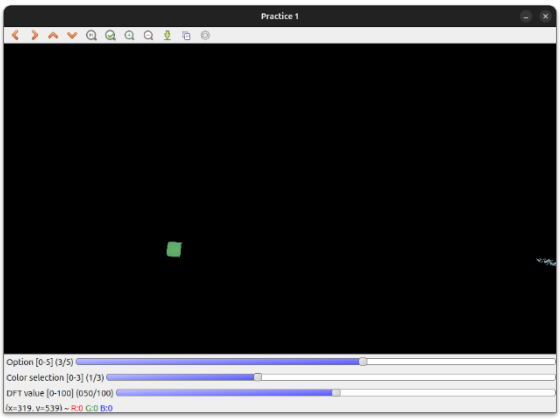
Funciones del Trackbar, waitKey, pollKey:
https://docs.opencv.org/3.4/d7/dfc/group__highgui.html

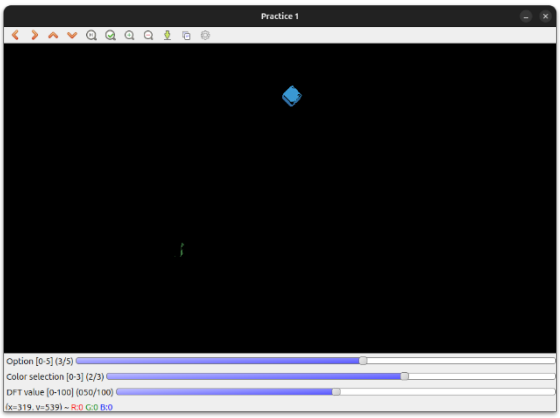
Imágenes

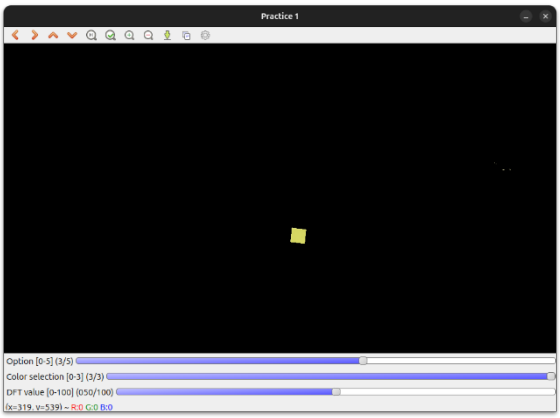

Practice 1
Option [0-5] (1/5)
Color selection [0-3] (0/3)
DFT value [0-100] (050/100)

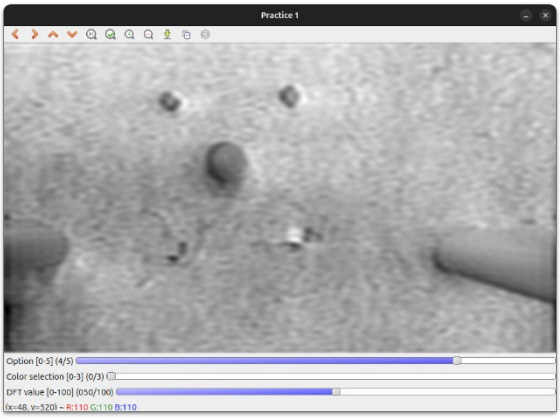

Practice 1
Option [0-5] (2/5)
Color selection [0-3] (0/3)
DFT value [0-100] (050/100)

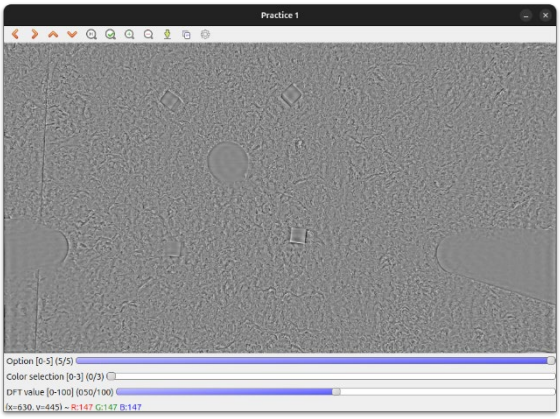

Practice 1
Option [0-5] (3/5)
Color selection [0-3] (0/3)
DFT value [0-100] (050/100)


Practice 1
Option [0-5] (3/5)
Color selection [0-3] (1/3)
DFT value [0-100] (050/100)


Practice 1
Option [0-5] (3/5)
Color selection [0-3] (2/3)
DFT value [0-100] (050/100)


Practice 1
Option [0-5] (3/5)
Color selection [0-3] (3/3)
DFT value [0-100] (050/100)


Practice 1
Option [0-5] (4/5)
Color selection [0-3] (0/3)
DFT value [0-100] (050/100)


Practice 1
Option [0-5] (5/5)
Color selection [0-3] (0/3)
DFT value [0-100] (050/100)