

## Ejercicio 2 – Transformaciones del dominio y espacial

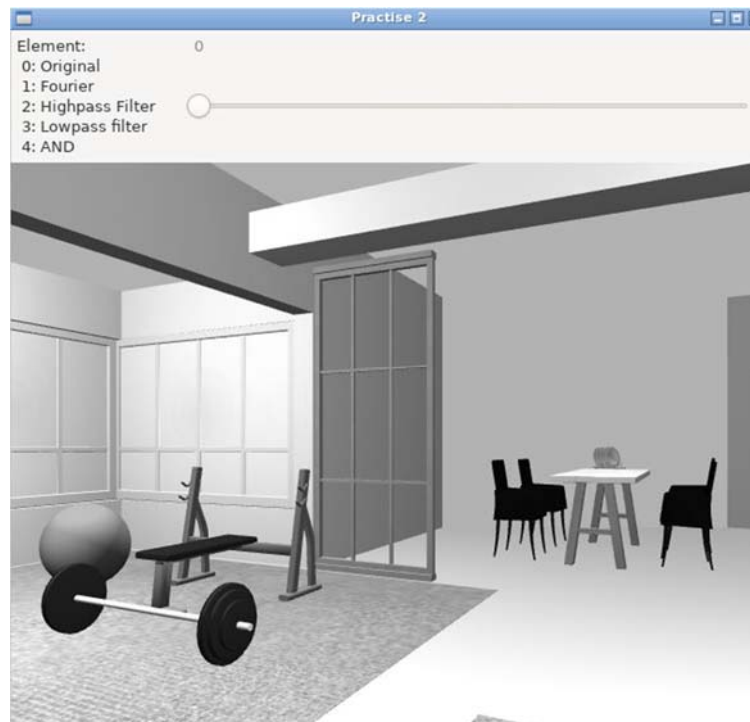
Este ejercicio tiene como objetivo aplicar los conceptos aprendidos en el Tema 3: Transformación del dominio y espacial.

La defensa del ejercicio se hará en clase, y hay que entregar un archivo **cv\_node\_p2.cpp** con el código generado deberás subir al Aula Virtual.

**Puntos totales posibles del ejercicio: 10**

### Instrucciones

Utilizando el simulador con Tiago, y partiendo del ejemplo **03\_01\_fourier**, se pide crear un programa que muestre en la parte superior un slider como el que muestra la figura:

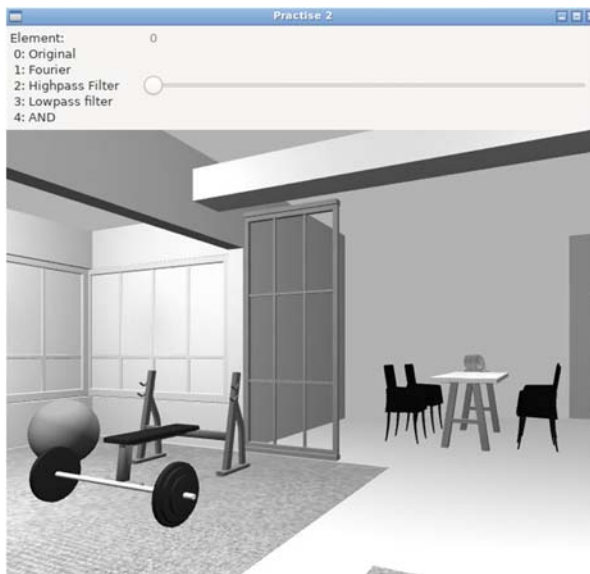


Dentro de la llamada al Callback del slider, o en una función aparte, se pide que en cada una de las 5 opciones se haga lo siguiente con la imagen **RGB** obtenida del Tiago:

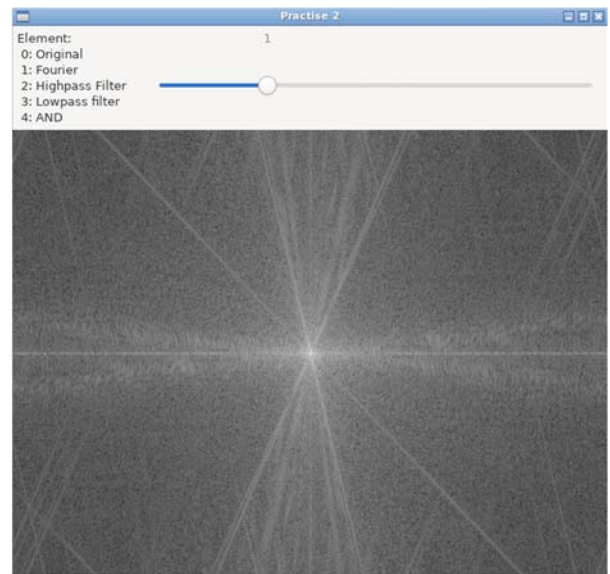
- Opción 0: Mostrar la imagen en formato de color GRAY.
- Opción 1: Mostrar la imagen del espectro de la transformada de Fourier.
- Opción 2: Aplicar un filtro paso alto al espectro de Fourier y mostrar la imagen inversa.
- Opción 3: Aplicar un filtro paso bajo al espectro de Fourier y mostrar la imagen inversa.

- Opción 4: Aplicar una transformación píxel a píxel con un operador de umbral y realizar la operación lógica AND. Para ello, se realizará una transformación de umbral de la imagen filtrada con un paso alto en el punto 2, y se le aplicará el criterio de umbralización de 0.4, 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 como paso bajo en el punto 3 y se le aplica el criterio anterior con un umbral de 0.6. Una vez obtenidas las dos imágenes, se realizará la transformación lógica AND.

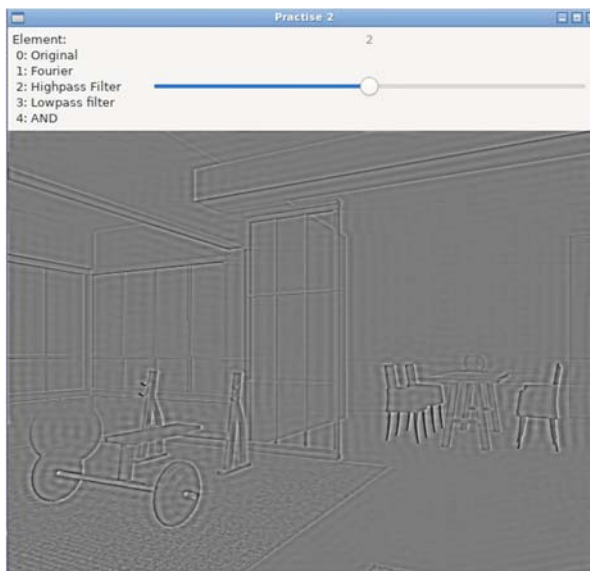
A continuación, se muestran unas capturas de lo que debe mostrar el resultado final para cada una de las opciones:



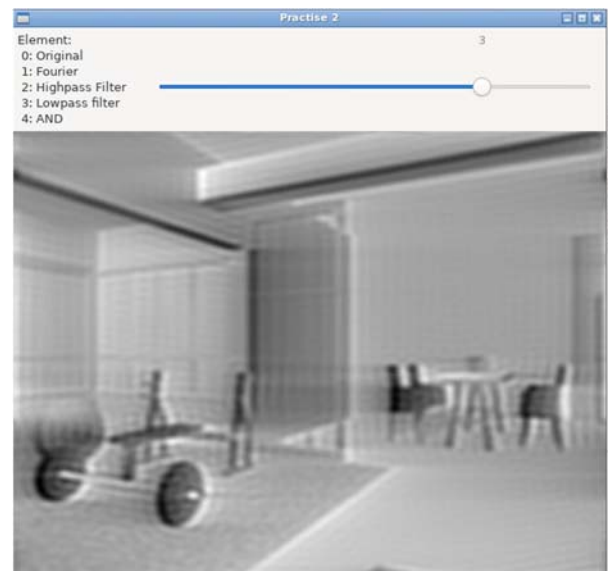
Opción 0



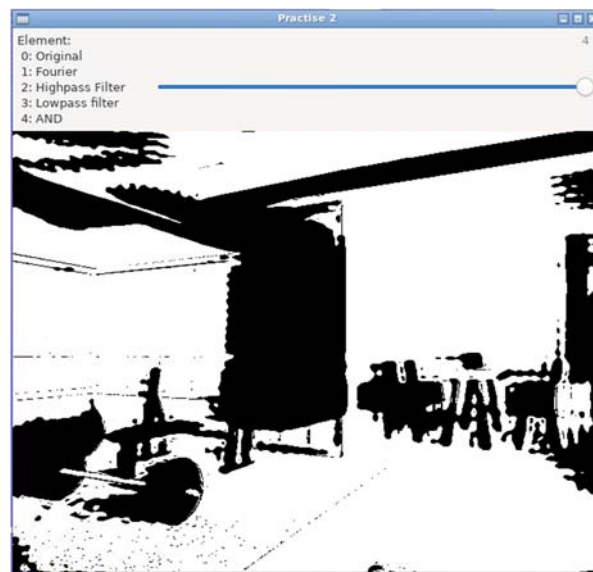
Opción 1



Opción 2

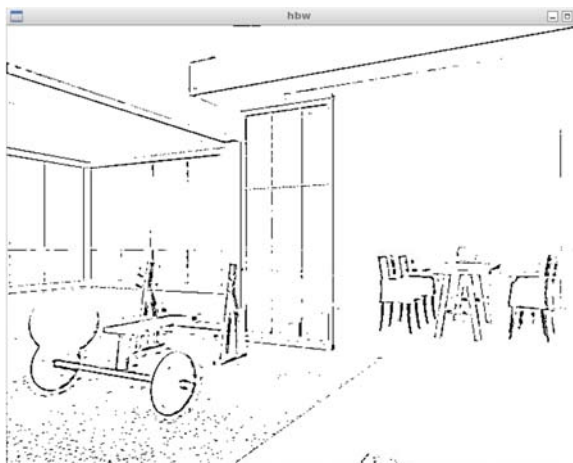


Opción 3



Opción 4

Esta última opción viene de aplicar la operación lógica AND sobre las siguientes imágenes, obtenidas de aplicar una transformación de píxel con un operador umbral:



Paso alto con umbral  $p = 0.4$



Paso bajo con 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 (paso alto o paso bajo), 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 dos formas distintas:

1. Generando una imagen de igual tamaño que el espectro, la cuál será blanca o negra dependiendo del filtro a aplicar, y dibujando un círculo cerrado centrado en el medio de la imagen de valor opuesto al fondo.
2. Mediante una función que genere una imagen de igual tamaño que el espectro, y dado un radio, mediante la distancia de cada uno de los píxeles con respecto al centro, si el píxel es menor que el radio se considera que está dentro del círculo, para lo cual se asignará un valor u otro dependiendo del filtro que se desee generar.

En cualquiera de los casos anteriores, el **radio del círculo** deberá ser de **50** para **ambos filtros**, y el resultado obtenido debe tener el siguiente aspecto antes de aplicar la multiplicación de los espectros:

