

## 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 que 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 texto como el que muestra la figura:



Se pide que en cada una de las 5 opciones se haga lo siguiente con la imagen **RGB** del Tiago:

- Opción 1: Mostrar la imagen en formato de **color GRAY**.
- Opción 2: Mostrar la imagen del espectro de la **transformada de Fourier**.
- Opción 3: Aplicar un filtro al espectro de Fourier que **mantenga** únicamente las frecuencias verticales y horizontales predominantes, y mostrar la imagen inversa.
- Opción 4: Aplicar un filtro al espectro de Fourier que **elimine** las frecuencias verticales y horizontales predominantes, y mostrar la imagen inversa.
- Opción 5: 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 en el **punto 3**, y 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 **punto 4** y se le aplica el criterio anterior con un **umbral de 0.4**. Una vez obtenidas las dos imágenes, se realizará la transformación lógica AND.

## Visión Artificial

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



Opción 1



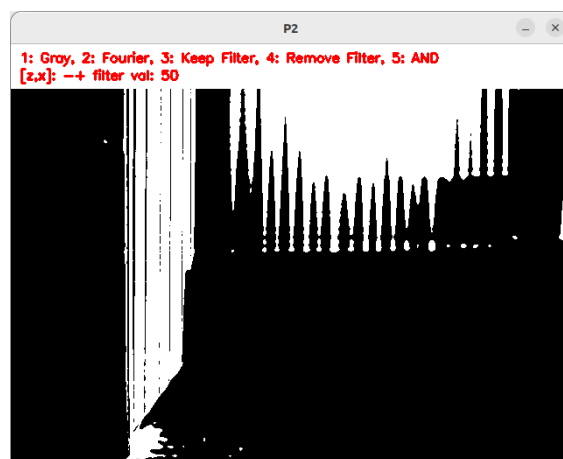
Opción 2



Opción 3

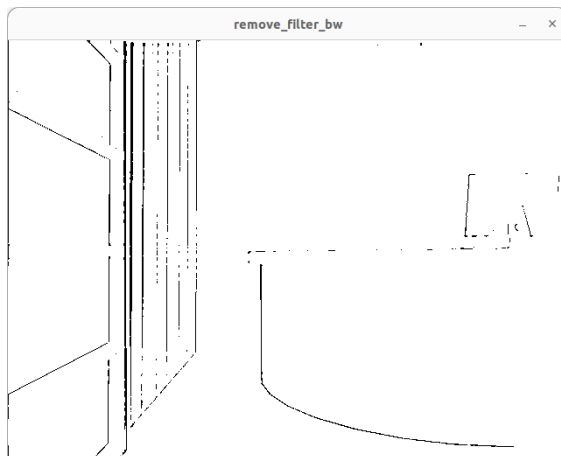


Opción 4

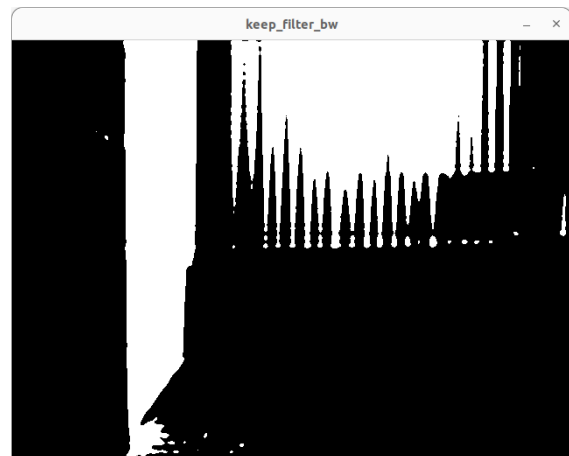


Opción 5

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:



Mantener frecuencias y umbral  $p = 0.4$



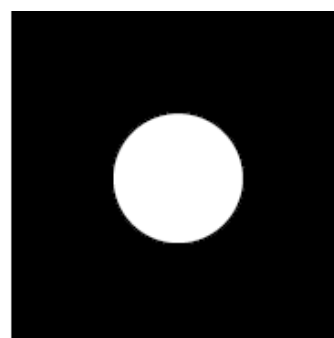
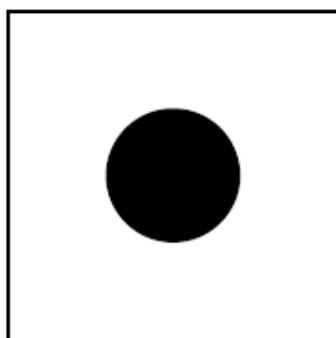
Eliminar frecuencias y umbral  $p = 0.6$

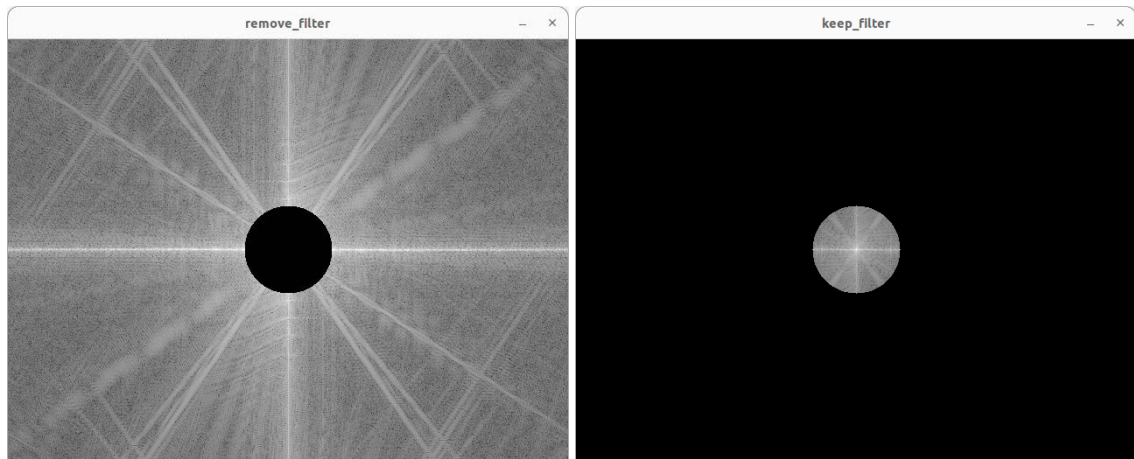
Se facilita la siguiente función, la cual se encarga de una vez obtenida la imagen con el filtro que se desea aplicar (mantener o eliminar), multiplica dicha imagen de filtro por la imagen del espectro de Fourier:

```
// 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 **50 a 100** para **ambos filtros**, y se modificará mediante las **teclas [z,x]** 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 las teclas [z,x] 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 5.

Se recomienda utilizar el mundo **aws\_bookstore** ya que contiene patrones que fácilmente son detectables.