

MasterFilter

Proyecto colaborativo entre Gustavo Magaña López y Salma Patricia Gutiérrez Rivera.

Universidad de Guanajuato

Campus León

División de Ciencias e ingenierías

Tarea : Funciones de Filtrado en Frecuencia

Materia : Fundamentos de procesamiento digital de imágenes.

Profesor : Dr. Arturo González Vega

Requerimientos y dependencias :

Preparar el entorno de ejecución usando conda

Para poder ejecutar el código de este repositorio, de la misma forma en la cual fue desarrollado, es necesario usar un ambiente virtual de Anaconda. Para más información, consultar:

1. Anaconda distribution.
2. conda environment

Para crear el ambiente virtual, asegúrese de tener la última versión de Anaconda instalada conda con el siguiente comando:

```
conda env create -f image_env.yml
```

Esto le preguntará si desea continuar, acepte tecleando 'y' en la sesión interactiva de la terminal. Habiendo creado el ambiente, para activarlo teclee el siguiente comando en su terminal:

```
conda activate image
```

Algunas de las dependencias (aquellas que no se encuentran en los repositorios de Anaconda) fueron instaladas usando `pip`. Nótese que dicho `pip` no es el que tiene su instalación principal de Python. **Asegúrese de activar el ambiente con `conda activate image`** antes de ejecutar el siguiente comando.

```
pip install -r requirements.txt
```

Ahora está listo para correr todos los scripts aquí contenidos.

Indicaciones y respuestas de la tarea :

Fundamentos de procesamiento digital de imágenes : Funciones de filtrado en frecuencia

1. (2 puntos) Modificar las funciones vistas en clase que generan filtros de tipo ideal de la forma pasa-bajos, pasa-altos, pasa-bandas rechazo de bandas, generalizandolas para que se indique si el filtro es ideal, Gaussiano o de Butterworth
 - En el archivo `mfilt_funcs.py`, las funciones que generan los filtros son:
 - `kernel_lowpass()`
 - `kernel_highpass()`
 - `kernel_band_reject()`
 - `kernel_band_pass()`
 2. (3 puntos) Hacer una función que diseñe filtros notch ideales, de Gauss o de Butterworth.
 - En el archivo `mfilt_funcs.py`, la función que genera los filtros notch es:
 - `kernel_notch()`
 3. (1 punto) Hacer una función maestra que diseñe filtros indicando: Tipo de filtro (pasa bajos, pasa altos, pasa bandas, rechazo de bandas, notch), con que forma del filtro (ideal, Gaussiano, Butterworth) y los parámetros necesarios para realizar el diseño.
 - En el archivo `mfilt_funcs.py`, la función maestra que diseña cualquiera de los filtros anteriormente mencionados es:
 - `master_kernel()`
 4. (1 punto) Una función que diseñe y aplique el filtro a una imagen.
 - En el archivo `mfilt_funcs.py`, la función que diseña y aplica los filtros es:
 - `filtra_maestra()`
 5. Reporte los siguientes resultados:
 - Véase el archivo “Filtro_maestro.pdf”
- (1 punto) A la imagen que se llama `mama.tiff`
- Filtro pasa bajos ideal con `wc=64`,
- Filtro pasa bajos butt con `wc=64`, `orden=2`
- Filtro pasa bajos gauss con `wc=64`,
- Filtro pasa altos gauss con `wc=64`,

Filtro pasa bandas gauss con $wc1=54$, $wc2=74$,

Filtro rechazo de bandas gauss con $wc1=54$, $wc2=74$,

6. (2 puntos) A las figuras FigP0405(HeadCT_corrupted).tif y a RadiografiaRuidoCoherente.jpg intente quitar el ruido coherente que se observa, detalle el procedimiento que realizó y los resultados que obtuvo