

1. Objetivos

Este Práctico de Laboratorio (PdL) tiene como objetivo general el desarrollo de técnicas de aplicación de herramientas matemáticas para tratamiento de señales sonoras y matriz de píxeles a fin de resolver el problema de filtrados en dominio de frecuencia. Dentre sus objetivos puntuales, destacan los siguientes.

- Implementación de código-fuente en Python que promueva la Transformada Discreta de Fourier (DFT) de una señal de audio estereo codificada como WAV o PCM;
- Análisis espectral de señal sonora;
- Implementación de código-fuente en Python que promueva la Transformada Discreta de Fourier (DFT) de una imagen de tipo PNG;
- Dimensionar un kernel de formato SINC (temporal) que cumple con un carácter pasa-banda del ancho de frecuencias $200Hz$ y $3kHz$ para filtrar una señal sonora;
- Dimensionar un sistema de detección de bordes de una imagen en escala de gris;
- Dimensionar un sistema de suavización de una imagen en escala de gris.

2. Material

- Computadora con Python 3.10 (o superior) e IDE/editor de código;
- Librerías Python `matplotlib.pyplot`, `numpy` y `opencv`.

3. Metodología

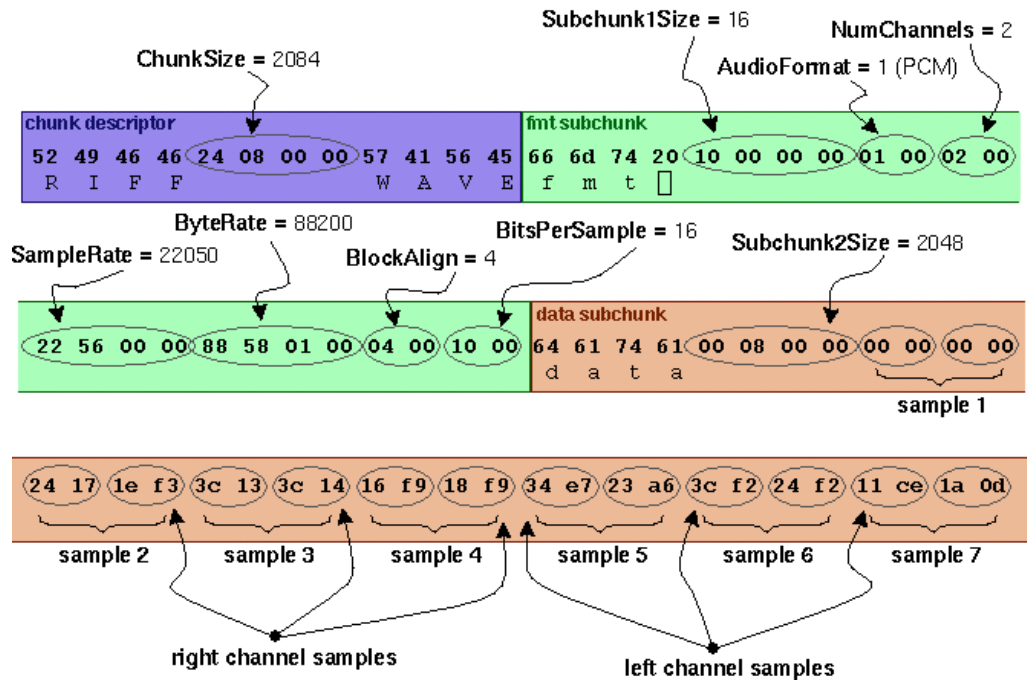
Considerar lo desarrollado en el PdL I. Agregar para la clase **Senal**, previamente implementada, las siguientes funcionalidades.

3.1. Filtrado de señal sonora

Para este apartado, considerar el archivo adjunto `violin_cut.wav` como la señal sonora a ser manipulada y promover los siguientes cambios.

Recordar que la configuración de un archivo WAV está configurado con metadatos, tal cual ilustra la Figura 1.

Figura 1: Estructura de archivo WAV estereo y metadatos.



Fuente: WAVE PCM soundfile format; [4]

3.1.1. Transformada de Fourier

Agregar a la clase **Senal** el método **Senal.DFT()** que promueve la DFT de su señal temporal.

Además, también se deberá implementar el método que promueve la transformada inversa, **Senal.iDFT()**.

3.1.2. Análisis espectral

Luego de configurado las gráficas de segmentos de la señal en dominio de frecuencia, identificar cuales son las notas musicales más significantes del sonido. Es decir, las más altas amplitudes en dominio de frecuencia deben tener una relación con la escala de notas musicales, como muestra la Figura 2.

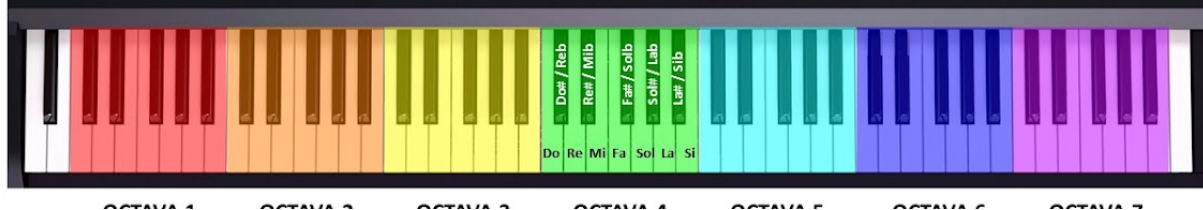
3.1.3. Kernel pasa-banda

Como establecido el dimensionamiento de un kernel en [1], a través de su Cap. 16, *Windowed-Sinc Filters*, utilizar de las frecuencias de corte 200Hz (menor) y 3kHz (mayor) para el diseño de un filtro para generar el efecto de pasa-banda que considera este *spectrum*.

Para eso, se debe recordar de la metodología de creación de un filtro digital, como ha sido mencionado en la clase PSAD_07.audio.PDF.

Figura 2: Escala de frecuencias de las notas musicales.

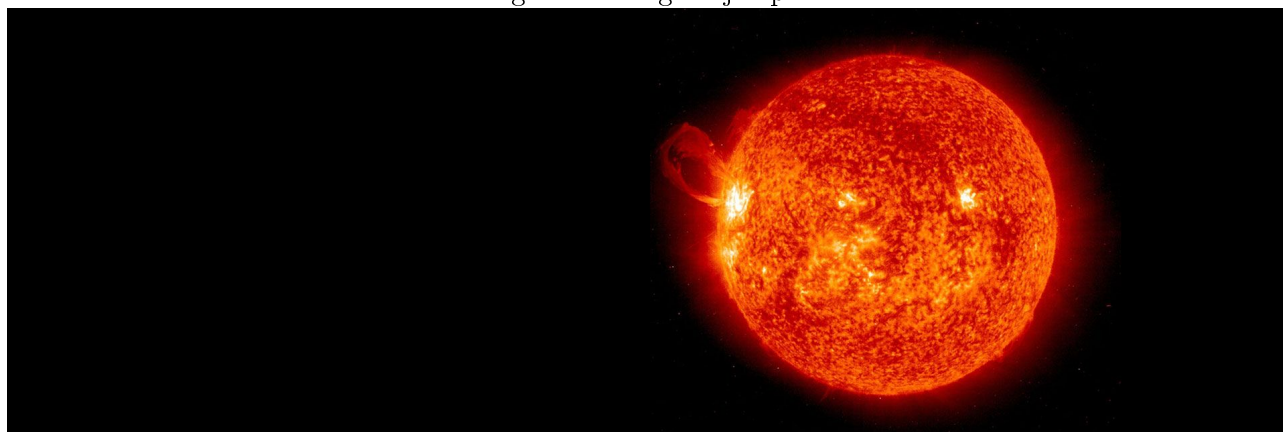
FRECUENCIA DE LAS NOTAS MUSICALES EN HERCIOS (Hz)									
	OCTAVA 0	OCTAVA 1	OCTAVA 2	OCTAVA 3	OCTAVA 4	OCTAVA 5	OCTAVA 6	OCTAVA 7	OCTAVA 8
Do	16,3516	32,7032	65,4064	130,813	261,626	523,251	1046,50	2093,00	4186,01
Do# / Reb	17,3239	34,6479	69,2957	138,591	277,183	554,365	1108,73	2217,46	4434,92
Re	18,3540	36,7081	73,4162	146,832	293,665	587,330	1174,66	2349,32	4698,64
Re# / Mib	19,4454	38,8909	77,7817	155,563	311,127	622,254	1244,51	2489,02	4978,04
Mi	20,6017	41,2035	82,4069	164,814	329,628	659,255	1318,51	2637,02	5274,04
Fa	21,8268	43,6536	87,3071	174,614	349,228	698,456	1396,91	2793,83	5587,66
Fa# / Solb	23,1246	46,2493	92,4986	184,997	369,994	739,989	1479,98	2959,96	5919,92
Sol	24,4997	48,9995	97,9989	195,998	391,995	783,991	1567,98	3135,96	6271,92
Sol# / Lab	25,9565	51,9130	103,826	207,652	415,305	830,609	1661,22	3322,44	6644,88
La	27,5000	55,0000	110,000	220,000	440,000	880,000	1760,00	3520,00	7040,00
La# / Sib	29,1353	58,2705	116,541	233,082	466,164	932,328	1864,66	3729,31	7458,62
Si	30,8677	61,7354	123,471	246,942	493,883	987,767	1975,53	3951,07	7902,14



3.2. Manipulación de imagen

Para este apartado, considerar el archivo adjunto `sun.jpg` como la señal de entrada a ser manipulada y promover los siguientes cambios.

Figura 3: Imagen ejemplo.



3.2.1. Suavizado

Para la imagen de entrada, promover la creación de una máscara gaussiana para el suavizado a fin de eliminar el ruido, considerando lo mencionado en la clase `PSAD_08.imagenes.PDF`.

3.2.2. Detección de bordes

Para la imagen de entrada, promover la detectar los bordes de la imagen a fin de poder concluir que hay una zona de interés cerrada (el círculo que define el sol) y dibujar un rectángulo o un círculo en esa zona.

Para identificar la zona de interés, se puede aplicar una verificación en donde el contraste es significativo.

4. Resultados

Realizar un informe relatando todos los procedimientos, resultados y dificultades en el práctico y entregarlo, hasta la fecha indicada, en formato PDF en adjunto con el código-fuente PY por vía de la tarea asignada en Moodle. El informe deberá contener la descripción de las metodologías utilizadas, los gráficos extraídos de la ejecución del programa como resultados obtenidos por medio del módulo `matplotlib.pyplot.savefig()` y el código-fuente implementado como apéndice.

Para la realización del informe, se debe obedecer los formatos \LaTeX , ODT o DOCX que están disponible en Moodle.

5. Conclusiones

Descripción de las conclusiones y de lo que se puede y no se puede inferir de los resultados obtenidos presentados en el Apartado 4. Dentre estas inferencias, se pueden responder a cuestiones como las siguientes.

Referencias

- [1] SMITH, S., *The scientist and engineer's guide to digital signal processing*. California Technical Publishing. 1999. ISBN 0-9660176-6-8. www.dspguide.com
- [2] DOWNEY, A., *Think Python: how to think like a computer's scientist*, versión 2.0.17. Green Tea Press. 2012.
- [3] DOWNEY, A., *Think DSP: digital signal processing in Python*, versión 1.0.9. Green Tea Press. 2014.
- [4] SAPP, C., *WAVE PCM soundfile format*. v2.1.0. 2004. <http://soundfile.sapp.org/doc/WaveFormat/>