UNIVERSIDAD SERGIO ARBOLEDA

Análisis de señales

Laboratorio 03: Introducción a Matlab: imágenes, audio y vídeo

Escuela de Ciencias exactas e Ingeniería

Código: SA2020IIG02_LAB03

Profesor: Marco Teran

Name: _____ **Deadline:** 15 de octubre 2020

Abstract

En el presente laboratorio se aplicarán las bases para desarrollar los posteriores laboratorios, talleres y trabajos escritos. Se realizará código de MATLAB que permita la generación, tratamiento y almacenamiento de imágenes, audio y vídeo. Se compilarán códigos en LaTeXy se generarán archivos *PDF* de acuerdo al formato de presentación. En el presente laboratorio se implementarán algunas herramientas necesarias para el procesamiento de archivos *multimedia* en MATLAB. Se realizarán códigos en MATLAB para la generación, transformación y almacenamiento de archivos de audio e imágenes. Se abrirán en MATLAB archivos de video y se obtendrá información de estos.

1 Desarrollo de la práctica de laboratorio

1. (5 points) **Responda brevemente en la sección de Marco Teórico de su plantilla de laboratorio las siguientes preguntas:**

(a) En el marco teórico es necesario definir las características, tipo de codificación y propiedades de un archivo de audio con extensión *.wav*.

(b) ¿Que diferencia hay entre archivos de audio *mono* y *estéreo* al ser leídos y guardados en MATLAB?

(c) Para leer una imagen en Matlab es necesario conocer el tipo de formato que maneja la imagen y su codificación. Describa los formatos *.PNG*, *.BMP* y *.JPG*, diferencie, preferiblemente con una tabla. ¿Que son los modelos de color RGB y CMYK?

2. (10 points) **Exportar señales de MATLAB a archivos *.wav*:**

Un archivo *.wav* es un estándar de audio para computadores[1].
A continuación se muestran los comandos utilizados para abrir, leer y guardar archivos *.wav* en Matlab.
La linea de comando
[x, fs, bits] = **wavread**('filename')
Lee un archivo llamado *'filename.wav'* y lo convierte en el vector x como variable de Matlab; fs extrae la frecuencia de muestreo del archivo y *bits*, tal cual como su nombre lo indica, representa la resolución de bits de la señal muestrada. El comando
x = **wavread**('filename',Nsamples)
lee las primeros $Nsamples$ muestras del archivo *.wav* en caso de que no se desee extraer el audio completo.
El comando
wavwrite(x,fs, 'filename')
crea a partir del vector x un archivo de audio *.wav* con una frecuencia de muestreo fs y llamado *'filename.wav'*.
La resolución estándar para archivos de audio es de 16 *bits* o ± 32768 niveles. La amplitud se encuentra escalada y restringida a un intervalo de $[-1, 1]$ en el vector x. El comando
wavwrite(x,Fs,bit, 'filename')
cambia la resolución de bits del archivo.
A continuación genere en MATLAB una señal sinusoide con una amplitud de $A = 0.1$, una frecuencia

[1] WAV, articulo *Wikipedia* https://en.wikipedia.org/wiki/WAV

$f = 100\,Hz$, para $N = 100000$ muestras, y una frecuencia de muestreo de $f_s = 22050\,Hz$.
Escuche la señal, y luego guarde esta en un archivo *.wav*, cuyo nombre de archivo es el código asignado al laboratorio. Adjunte el archivo *.wav* en el proyecto *.zip* de su laboratorio.
El tiempo total de la señal: $N_{samples}/fs = 4.535\,s$.

```
N_samples = 100000;
fs = 22050;
fc = 100;
t = (0:N-1)/fs;
x = 0.1*sin(2*pi*fc*t);
sound(x,fs)
wavwrite(x,fs,'sinsound.wav');
```

Código 1 – Generación de una señal *.wav* a partir de un armónico

Genere y guarde una versión ruidosa de esta señal, es decir adicione ruido blanco Gaussiano con una potencia igual a la mitad de la potencia de la señal de audio original. Llame esta versión del audio contaminado tal cual como la anterior, pero agregue sin espacios al final del nombre la palabra *NOISED*.

3. (10 points) **Tratamiento de imágenes en Matlab:**

Para abrir la imagen lena.png, de tipo RGB que se encuentra en el repositorio, se debe utilizar la función:

RGB = imread('Lena.png');

En esta linea de código anterior, la imagen[2] se guarda en una variable-vector denominado RGB. Describa las dimensiones de la variable RGB y su significado. ¿Que tipo de valores toma el vector? Defina la estructura, y por ultimo genere las gráficas de los 3 histogramas correspondientes a los datos de las tres matrices. Utilice las funciones especializadas de Matlab de DIP (*ing.* Digital Image Processing) si así lo desea.

Para mostrar los datos de la imagen RGB se utiliza la función imshow(RGB).
Para convertir y mostrar una una imagen a escala de grises utilicemos las siguientes lineas de comando:

```
gray = rgb2gray(RGB);
imshow(gray)
```

Código 2 – Exportar señales de MATLAB a archivos *.wav*

Describa las dimensiones de la variable *gray*. ¿Que tipo de valores toma? Defina su estructura, y genere un histograma con sus valores.

[2]Descargar la imagen de prueba de la carpeta */laboratory/introtoMatlab/pictures* del repositorio (Descargar).

El siguiente código extrae las componentes en rojo, verde y azul de la imagen peppers.png que se encuentra en la carpeta de introtoMatlab/pictures en el repositorio[3]:

```
close all; clear all; clc;
myimg=imread('peppers.png', 'PNG');
size(myimg)
nbcl=512; mcol=[0:nbcl-1]'/(nbcl-1);
mypal=zeros(nbcl,3,3); mypal(:,1,1)=mcol;
mypal(:,2,2)=mcol; mypal(:  ,3,3)=mcol;
for k=1:3
    figure(k), imagesc(myimg(:,:,k))
    colormap(mypal(:,:,k)); axis('image')
end
```

Código 3 – Extracción RGB de una imagen

1. Escriba una función código *.m* que lea una imagen, sea de color o en escala de grises, y la convierta en una imagen en blanco y negro (solo 1 *bit*). **Nota:** No utilice la función rgb2bw.
2. (**Investigación**) A una imagen en escala de grises agregue ruido blanco, de acuerdo a las siguientes relaciones señal ruido (SNR, *ing.* Signal to noise rate): $-4\,dB$, $0\,dB$, $2\,dB$. Muestre en *subplots* la imagen original, y las tres versiones ruidosas de esta.

4. (10 points) **Archivos de video en Matlab**

EL programa matemático Matlab puede leer distintos formatos de archivos de video como *.avi*, *.mpg* y *.wmv*. Para reproducir un video con MATLAB, basta con utilizar la linea de código
implay('rbsp_launch_720p.mp4');
Descargar el archivo de video aquí.

Para obtener la información de un archivo de video se puede utilizar el siguiente código:

```
movieObj = VideoReader('grail_launch_720p.wmv');
get(movieObj)
nFrames = movieObj.NumberOfFrames;
width = movieObj.Width;
height = movieObj.Height;
```

Código 4 – Extraer información de un archivo de video en MATLAB

Incluya en el informe la información del video.
Utilice el siguiente código para abrir, mirar la información de un archivo de video y mostrar de forma individual cada uno de sus *frames* (descargar archivo de video *oneCCC.wmv* que se encuentra en la carpeta introtoMatlab/video/ del repositorio del laboratorio enlace):

```
clear all
close all
movieObj = VideoReader('oneCCC.wmv');
get(movieObj)
nFrames = movieObj.NumberOfFrames;
for iFrame=1:2:nFrames
    I = read(movieObj,iFrame);
    fprintf('Frame %d\n', iFrame);
    imshow(I,[]);
    pause(0.1);
end
```

Código 5 – Abrir video en MATLAB y mirar sus frames

[3] Descargar peppers.png

Comente cada una de las lineas del código anterior e incluyalas en el informe, muestre los resultados.

Para más ejemplos de código Matlab utilizando videos, visitar:
http://www.mathworks.com/help/matlab/ref/videoreader.read.html