

### Resumen

En el presente laboratorio se realizará código de MATLAB que permita la generación, tratamiento y almacenamiento de imágenes, audio y vídeo. 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 vídeo y se obtendrá información de estos. Se compilarán códigos en L<sup>A</sup>T<sub>E</sub>Xy se generarán archivos PDF de acuerdo al formato de presentación.

## 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<sup>1</sup>.

A continuación se muestran los comandos utilizados para abrir, leer y guardar archivos **.WAV** en MATLAB. La línea de comando

```
[x,fs,bits] = audioread('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 muestreada. El comando

```
x = audioread('filename',Nsamples)
```

lee las primeros **Nsamples** muestras del archivo **.WAV** en caso de que no se desee extraer el audio completo. El comando

```
audiowrite('filename',x,fs)
```

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  $\pm 32768$  niveles. La amplitud se encuentra escalada y restringida a un intervalo de  $[-1, 1]$  en el vector **x**. El comando

```
audiowrite('filename',x,Fs,bit)
```

cambia la resolución de bits del archivo.

A continuación genere en MATLAB una señal sinusoidal con una amplitud de  $A = 0,1$ , una frecuencia  $f = 100 \text{ Hz}$ , para  $N = 100000$  muestras, y una frecuencia de muestreo de  $f_s = 22050 \text{ Hz}$ .

Escuche la señal, y luego guarde esta en un archivo **.WAV**, cuyo nombre de archivo es el código asignado al laboratorio. Genere una gráfica y describa el archivo **.WAV** de audio generado en su documento de

<sup>1</sup>.WAV, artículo Wikipedia <https://en.wikipedia.org/wiki/WAV>

laboratorio.

El tiempo total de la señal:  $N_{samples}/fs = 4,535 s$ .

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

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 línea de código anterior, la imagen<sup>2</sup> 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 líneas de comando:

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

Código 2 – Convertir y mostrar una una imagen a escala de grises

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

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<sup>3</sup>:

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

Código 3 – Extracción RGB de una imagen

<sup>2</sup>Descargar la imagen de prueba de la carpeta `/laboratory/introtomatlab/pictures` del repositorio ([Descargar](#)).

<sup>3</sup>[Descargar peppers.PNG](#)

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):  $-4\text{ dB}$ ,  $0\text{ dB}$ ,  $2\text{ dB}$ . 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('rbasp_launch_720p.mp4');`

Descargar el archivo de video [aquí](#).

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

```
1 movieObj = VideoReader('grail_launch_720p.wmv');
2 get(movieObj)
3 nFrames = movieObj.NumberOfFrames;
4 width = movieObj.Width;
5 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](#)):

```
1 clear all
2 close all
3
4 movieObj = VideoReader('oneCCC.wmv');
5 get(movieObj)
6 nFrames = movieObj.NumberOfFrames;
7 for iFrame=1:2:nFrames
8     I = read(movieObj,iFrame);
9     fprintf('Frame %d\n', iFrame);
10    imshow(I,[]);
11    pause(0.1);
12 end
13
```

Código 5 – Abrir video en MATLAB y mirar sus frames

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

Para más ejemplos de código MATLAB utilizando videos, visitar:

<http://www.mathworks.com/help/matlab/ref/videoreader.read.html>

**NOTA:** Es obligatorio que se realice un *rigurosa* presentación de resultados obtenidos para cada sección del informe de laboratorio y se realice su respectivo análisis. Adicionalmente, no olvide incluir las conclusiones apropiadas del laboratorio.