

ITESM – CCM
Ingeniería Biomédica
Procesamiento y Análisis de Imágenes Médicas

PROCESAMIENTO DE IMAGENES EN MATLAB / SIMULINK

Pablo Ruiz López
A01335986

Martes, 22 Noviembre de 2016

Para la realización de la prueba final se desarrollaron 6 plataformas en la herramienta de MATLAB 2010 y Simulink. Cada uno de los archivos (.m y .mdl) esta ampliamente comentado y referenciado para explicar en cada paso el procedimiento.

La programación se desarrollo en un equipo de cómputo Apple® modelo MacBook Air con un procesador de 16 GHz doble Núcleo Intel i5 [4]; todos los programas desarrollados fueron funcionales en el equipo descrito y siguiendo las instrucciones del *Manual de Usuario* encontrado en **Anexo 1** de este documento. A continuación se muestran las descripciones de cada programa y los resultados obtenidos.

En el primer programa '*ExFinalPreg1_1FiltroInv.mdl*' del examen (**Pregunta 1, inciso 1**) se incluye un escenario donde se realizan las siguientes tareas:

1.- Se lee la imagen de una hermosa mujer llamada Lena; a esta imagen se le hacen las conversiones de formato y de tipo de dato para poder operar con ella en forma matricial. Posteriormente se le agrega un filtrado basado en una convolución de la imagen y otra función matemática ($h(m,n)$, siendo m y n las dimensiones de la imagen); el resultado es que la imagen se ve alterada de tal forma que presenta un 'barrido'. No siendo suficiente, a la imagen se le agrega un tipo de ruido denominado 'sal y pimienta' ($w(m,n)$), el cual altera aún mas la imagen de tal forma que se encuentra barrida y con puntos negros en la pantalla ($y(m,n)$).

2.- Se realizan dos filtrados para cada una de las partes que generan distorsiones:

Un filtro basado en el cálculo del percentil para eliminar los puntos negros generados por el ruido 'sal y pimienta' y otro filtro basado en una convolución inversa con las mismas características que $h(m,n)$ pero operador opuesto. Con este proceso se logra restaurar la imagen ($x'(m,n)$). A continuación se muestra el escenario:

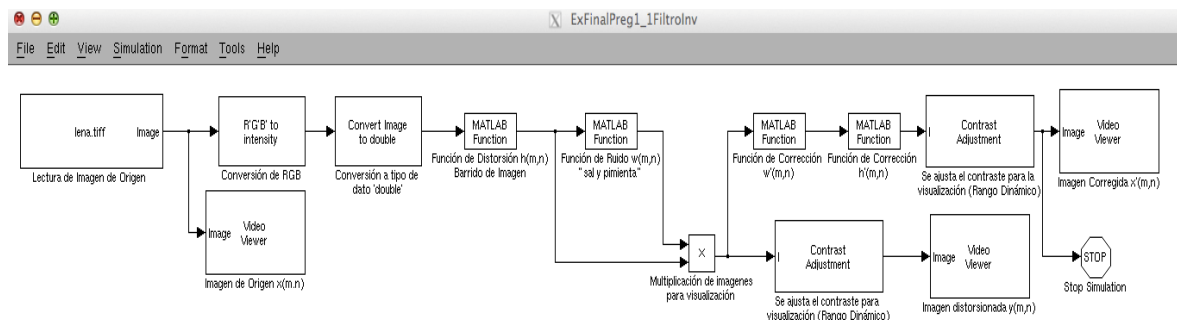


Fig. 1 Entorno gráfico de primer programa.

3.- Por ultimo se configura el contraste necesario y se incluyen los operadores para el despliegue correcto de las imágenes. El resultado es el despliegue de las 3 imágenes, Original ($x(m,n)$), Distorsionada ($y(m,n)$) y Recuperada ($x'(m,n)$):

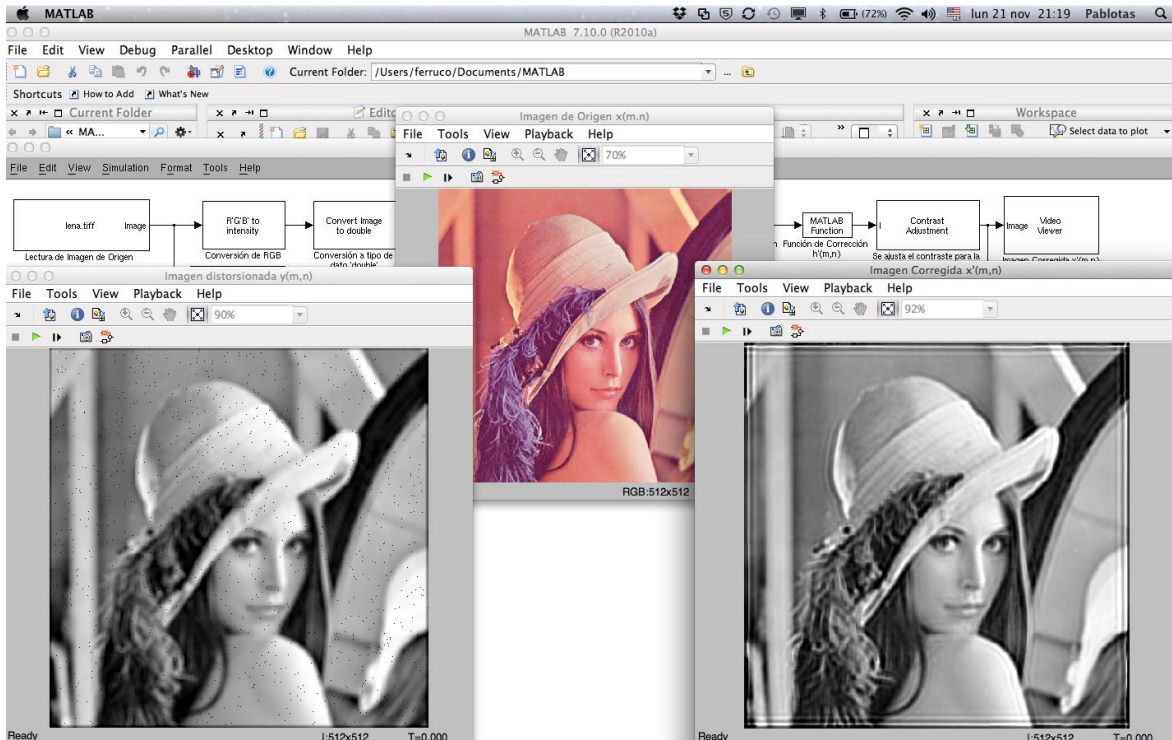


Fig. 2 Resultados obtenidos en primer programa.

En el segundo programa **'ExFinalPreg1_2FiltroWei.mdl'** del examen (**Pregunta 1, inciso 2**) se incluye un escenario donde se realizan las siguientes tareas:

- 1.- Se lee la imagen de una hermosa mujer llamada Lena; a esta imagen se le hacen las conversiones de formato y de tipo de dato para poder operar con ella en forma matricial. Posteriormente se le agrega un filtrado tipo 'motion' ($h(m,n)$); el resultado es que la imagen presenta un 'barrido'. A la imagen se le agrega un tipo de ruido denominado 'auditivo' ($w(m,n)$), del cual se considera una varianza y un promedio cero, esta modificación altera aún mas la imagen ($y(m,n)$).
- 2.- Se realiza el filtrado tipo 'Weiner', del cual previamente se establecen los parámetros y las consideraciones necesarias como lo es un SNR (Radio de Señal a Ruido) igual a cero. Con este proceso se logra restaurar la imagen ($x'(m,n)$). A continuación se muestra el escenario:

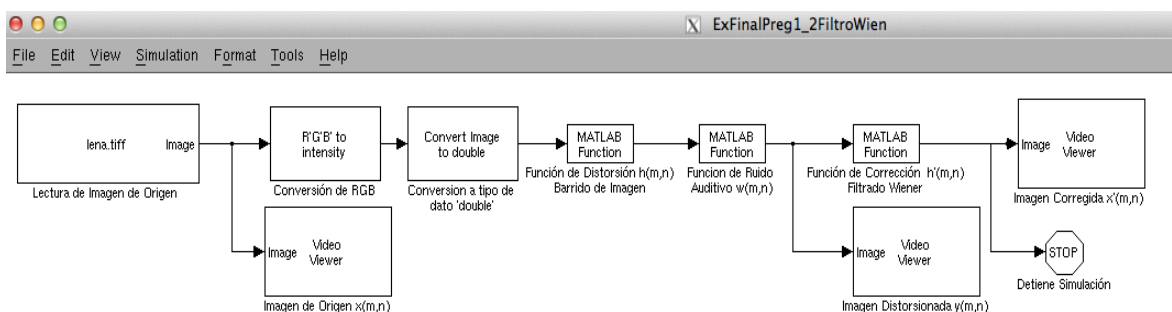


Fig. 3 Entorno gráfico del segundo programa.

3.- El resultado es el despliegue de las 3 imágenes, Original ($x(m,n)$), Distorsionada ($y(m,n)$) y Recuperada ($x'(m,n)$):

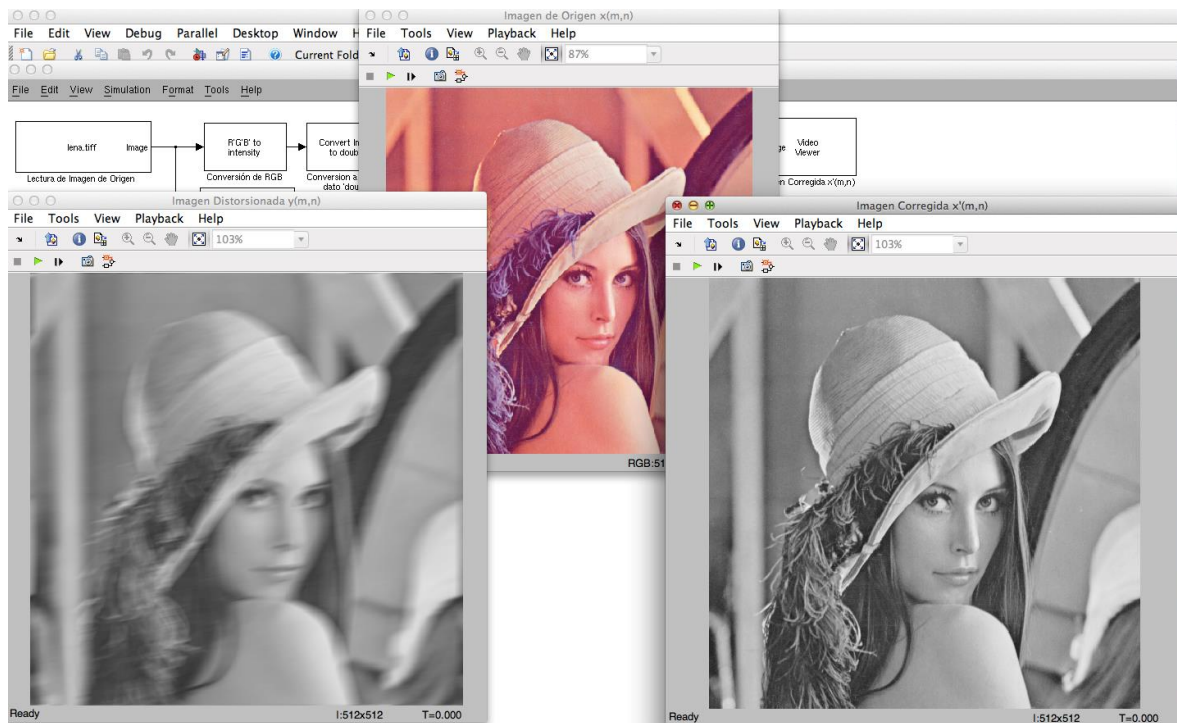


Fig. 4 Resultados obtenidos en el segundo programa.

En el tercer programa '*ExFinalPreg2.mdl*' del examen (**Pregunta 2**) se incluye un escenario donde se realizan las siguientes tareas:

- 1.- Se lee un video de unos robustos rinocerontes; a estas imágenes se les hacen las conversiones de formato y de tipo de dato para poder operar con ellas en forma matricial.
- 2.- A las imágenes se les hace la Transformación Cosenoidal Discreta (DCT), operación basada en el método de Transformada de Fourier (con la variante de nada más trabajar con números reales) en donde se llevan a las imágenes a un cambio de dimensiones basadas en coeficientes cosenoidales en donde se puede operar con información relevante de una manera más sencilla. Esta transformación se realiza para la compresión de las imágenes (estándar creado en los años 80 cuyo nombre formal es ISO/IEC 10918-1 y que ahora es ampliamente utilizado en compresiones JPEG, por ejemplo).

Los pasos para realizar la compresión son: transformación al espacio dimensional, división de la imagen en matrices de 16x16 (parámetro especificado para este ejercicio), transformación cosenoidal, cuantificación y lectura de los bloques.

- 3.- El resultado es el despliegue de 2 videos: el video original y el video comprimido; a continuación se muestran dichos resultados y el diagrama de bloques en Simulink:

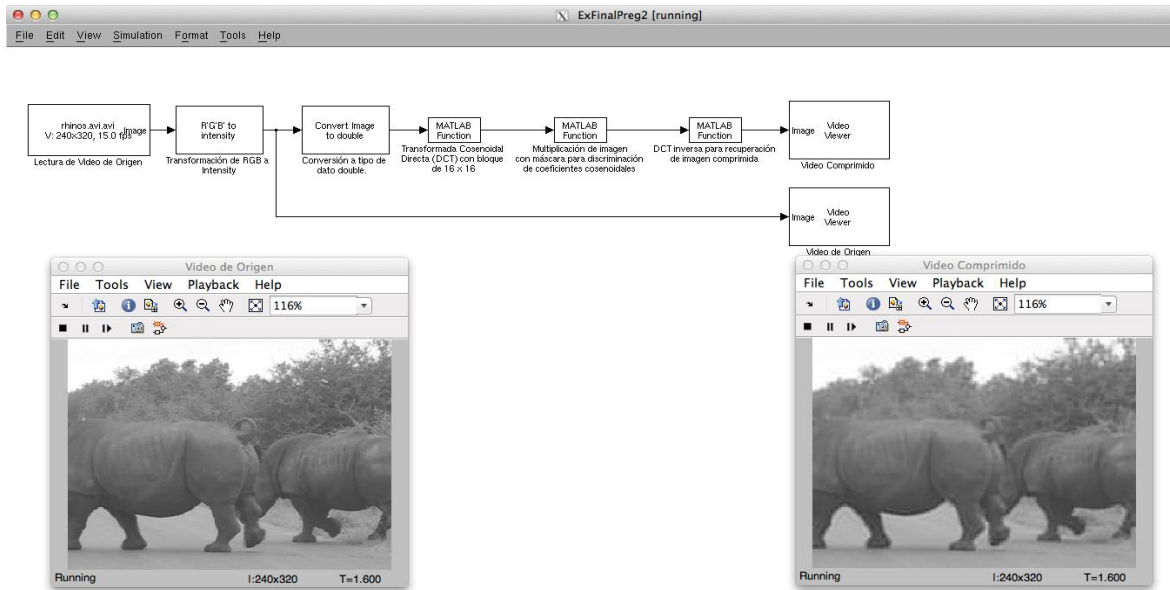


Fig. 5 Entorno gráfico y resultados obtenidos en el tercer programa.

Discusión:

Como se puede observar en las dos imágenes, la calidad difiere entre una y otra es distinta y esto es debido a la pérdida de información en el proceso de compresión.

La DCT es una técnica que divide la señal discreta de la imagen en varias señales discretas derivadas de la señal original; estos coeficientes marcan el espacio a donde se va a llevar la imagen y poder operar con ella en este dominio en donde se logre eliminar la información redundante o imperceptible al ojo humano.

La el tipo de información eliminada en la imagen a comprimir determina si esta siendo comprimida mediante técnicas de **compresión sin pérdidas** (las cuales consisten en la eliminación de redundancia estadística) o **compresión con pérdidas** (la cual altera la imagen quitando la información 'imperceptible' y aprovechando también la eliminación de redundancia estadística). La metodología DCT pertenece a la categoría de compresión con pérdidas, es por ello que se logra recuperar un porcentaje alto de la imagen pero no del todo reconstruida al 100% como lo sería una compresión sin pérdidas.

En el cuarto y quinto programa '*ExFinalPreg3_1.mdl*' y '*ExFinalPreg3_2.mdl*' del examen (**Pregunta 3**) se incluyen escenarios donde se realizan las siguientes tareas:

1.- Se lee la imagen de un grupo de valientes jóvenes después función de baile (Imagen A); a esta imagen se le hacen las conversiones de formato y de tipo de dato para poder operar con ella en forma matricial. Posteriormente se leen dos imágenes del rostro de 2 de los bailarines (Imagen B y C) a las cuales también se les hacen las conversiones de formato y de tipo de dato.

2.- Etiquetado: posteriormente se realiza la operación de correlacionar la Imagen A con las imágenes B y C, de tal forma que se van a comparar estas imágenes y se obtendrán datos estadísticos basados en los datos de las imágenes en operación (el tamaño de las imágenes B y C debe de ser menor que el de la imagen A para poder 'recorrer' B y C sobre A y comparar por zonas). Una vez haciendo esta comparación se agrega el operador que identifica en donde la varianza de los datos es cero o es mínima; en esa área se va a dibujar un rectángulo que indicara la aproximación del sistema (para la correcta colocación de este rectángulo dibujado en color negro se agregan operadores que modifican la posición de manera simétrica a los tamaños de las imágenes). A continuación se muestra el escenario:

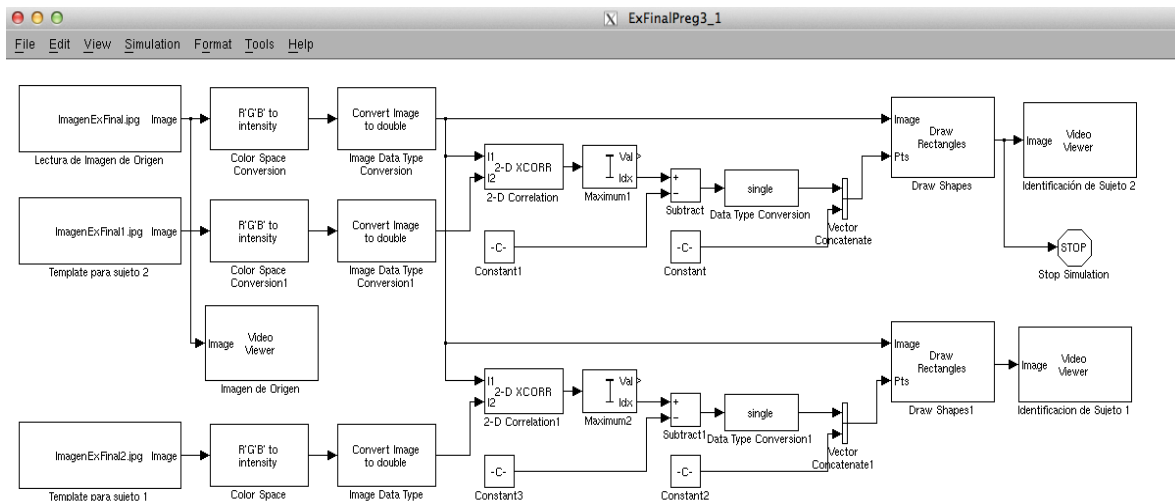
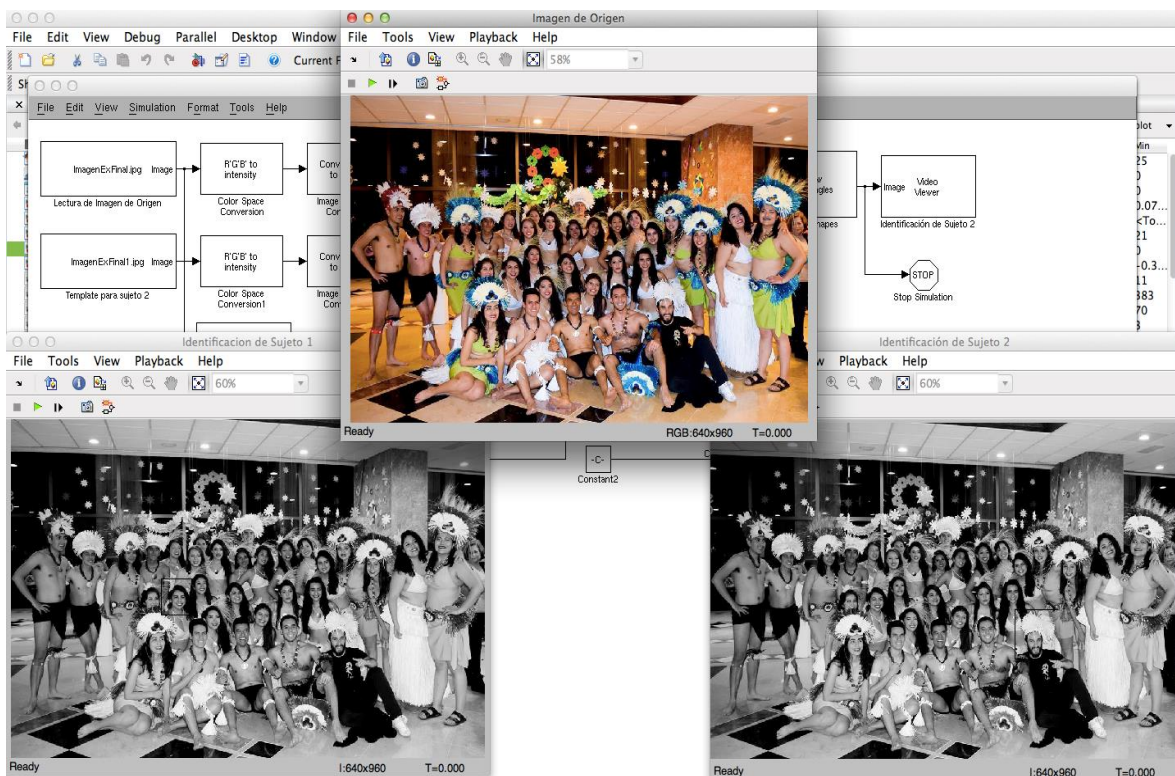


Fig. 3 Entorno gráfico del cuarto y quinto programa (con la variación en el tipo de dato en las cajas 'Lectura de Imagen de Origen' y 'Template para sujeto 1 y 2'.)

3.- El resultado es el despliegue de 3 imágenes: Imagen A, Imagen B e Imagen C. En las dos últimas se etiqueta a un apuesto sujeto en particular. A continuación se muestra como es que se despliegan los resultados en las imágenes:

Despliegue de imágenes:



Sujetos identificados:



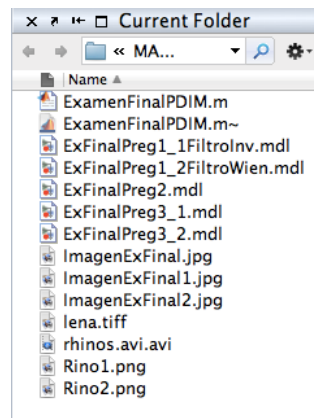
Para el quinto programa *'ExFinalPreg3_2.mdl'* se realizan los mismos pasos descritos anteriormente, con la variación de en vez de identificar a dos sujetos en una foto de bailarines, se identifica a los dos robustos rinocerontes en un video mostrado (como en el tercer programa). Cabe señalar que este programa en particular puede tomar tiempo excesivos en correr e ir reproduciendo el video.

Anexo 1. Manual de Usuario

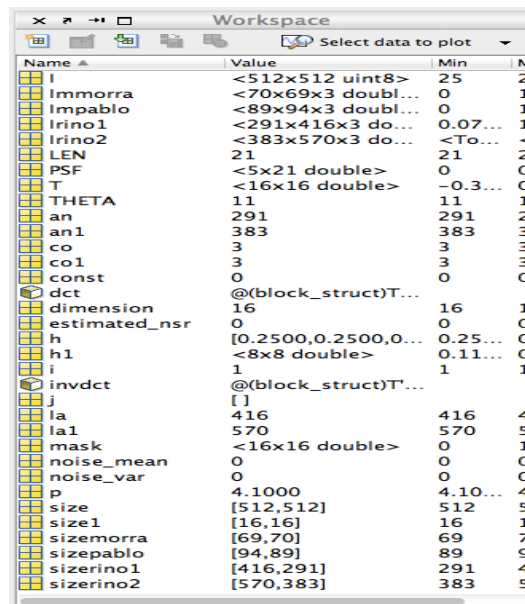
En la carpeta *ExFinal_PabloRuizL.zip* se encuentran todos los archivos que se realizaron para la prueba final, en ellos se encuentran 5 programas de Simulink (.mdl); 1 archivo de MATLAB (.m) con las variables necesarias; las imágenes y videos a utilizar en cada escenario: 2 (.jpg), 2 (.png), 1 (.tiff) y 1 (.avi); de igual forma se encuentra un archivo pdf con la descripción de la prueba y este manual de uso.

Pasos a seguir para la evaluación de los programas

- 1.- Descomprimir la carpeta *ExFinal_PabloRuizL.zip*
- 2.- Cargar los archivos que se encuentren en el archivo descomprimido a la carpeta de trabajo utilizada para MATLAB.



3.- **IMPORTANTE:** Correr el archivo *ExamenFinalPDIM.m* (asegurar que en el 'workspace' se generen las variables declaradas en este archivo, de lo contrario los programas en Simulink no tendrán los parámetros y archivos necesarios para funcionar y arrojarán errores).



Name	Value	Min	Max
I	<512x512 uint8>	25	255
Imorra	<70x69x3 double>	0	1
Impablo	<89x94x3 double>	0	1
Irino1	<291x416x3 double>	0.07...	1
Irino2	<383x570x3 double>	<To...	1
LEN	21	21	21
PSF	<5x21 double>	0	1
T	<16x16 double>	-0.3...	1
THETA	11	11	11
an	291	291	291
an1	383	383	383
co	3	3	3
co1	3	3	3
const	0	0	0
dct	@(block_struct)T...		
dimension	16	16	16
estimated_nsr	0	0	0
h	[0.2500,0.2500,0...	0.25...	0.25...
h1	<8x8 double>	0.11...	1
i	1	1	1
invdct	@(block_struct)T...		
j	[]		
la	416	416	416
la1	570	570	570
mask	<16x16 double>	0	1
noise_mean	0	0	0
noise_var	0	0	0
p	4.1000	4.10...	4.10...
size	[512,512]	512	512
size1	[16,16]	16	16
sizemorra	[69,70]	69	69
sizemorra	[94,89]	89	89
sizerino1	[416,291]	291	291
sizerino2	[570,383]	383	383

4.- Abrir los archivos .mdl e iniciar la simulación en cada uno de ellos (**NOTA:** el orden en el que se vayan evaluando los programas no es influyente). Los archivos están guardados con el nombre siendo un indicador de las características a evaluar en cada pregunta del examen:

ExFinalPreg1_1FiltroInv.mdl = Pregunta 1, Filtro Invertido.

ExFinalPreg1_2FiltroWein.mdl = Pregunta 1, Filtro Weiner.

ExFinalPreg2.mdl = Pregunta 2, Compresión con CDT.

ExFinalPreg3_1.mdl = Pregunta 3, Identificación de 2 sujetos en foto.

ExFinalPreg3_2.mdl = Pregunta 3, Identificación de 2 sujetos en video.

Referencias:

[3] Compresión de Imágenes Optimizada en Consumo Energético Para Redes Inalámbricas, David Aledo Ortega, Universidad Politécnica de Madrid, POLITECNICA, Febrero de 2013.

[4] Especificaciones técnicas de laptop MacBook Air: <http://www.apple.com/macbook-air/specs.html>