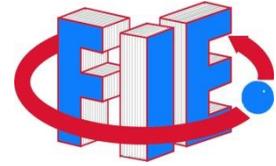




# UNIVERSIDAD MICHOACANA DE SAN NICOLAS DE HIDALGO



## Facultad de Ingeniería Eléctrica

### Laboratorio de Procesamiento Digital de Señales

#### PRÁCTICA 7

#### “Procesamiento de Imágenes”

---

#### Objetivo

Describir la información que conforma a una imagen digital y crear programa que sea capaz de realizar operaciones básicas con la imagen.

#### Antecedentes

##### Imagen Digital

Una imagen es la representación visual de un objeto o escena. Las imágenes digitales se obtienen a través de dispositivos de conversión analógico-digital incluidos en escáneres, cámara fotográfica digital, sistemas de radar, tomógrafos o directamente desde la computadora utilizando cualquier programa de tratamiento de imágenes.

De esta forma, la imagen obtenida es comunmente modelada como el resultado de una operación de filtro; la operación de filtrado, utilizando la teoría de espacio de señales puede realizarse como una multiplicación matricial:

$$\mathbf{u} = \mathbf{Sv}$$

donde  $\mathbf{u}$  es la imagen obtenida,  $\mathbf{S}$  es la matriz que representa al sistema y  $\mathbf{v}$  es la escena original.

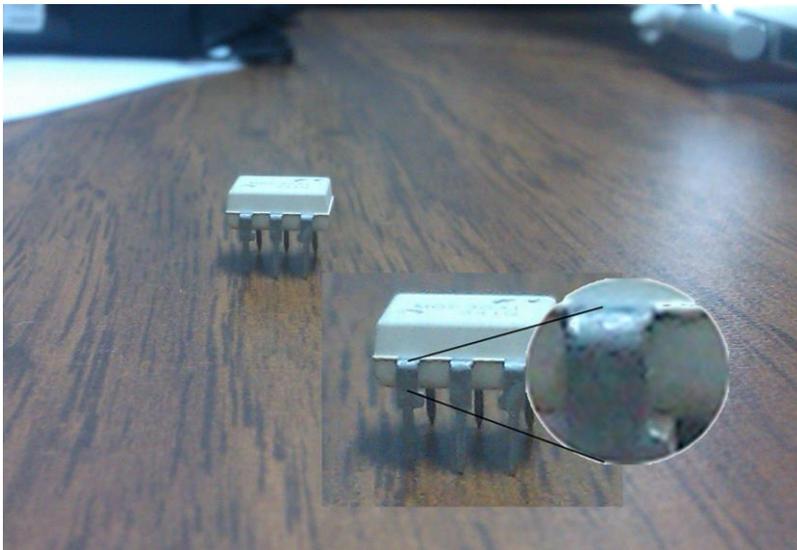
---

La información digital que genera cualquiera de los medios citados es almacenada en la computadora mediante bits (unos y ceros).

Las imágenes digitales se crean formando una retícula rectangular formada por celdillas. Cada una de las celdillas recibe el nombre de píxel.

Los píxeles representan una cantidad del fenómeno que mide el dispositivo de captura: en el caso de una fotografía, son las unidades de color que componen la imagen.

Un píxel no tiene una medida concreta, no se puede decir si mide 1 mm. o 1 cm. o 1 m., simplemente es la medida de división de una retícula en celdillas.



### **Profundidad de píxel**

Una imagen digital está formada por una matriz de píxeles ( $a \times b \times c$ ), donde  $a$  y  $b$  representan la anchura y altura y  $c$  es la profundidad de color o profundidad de bit, podríamos decir que es la tercera dimensión de la matriz, la que permite que cada píxel pueda tener un número determinado de colores distintos.

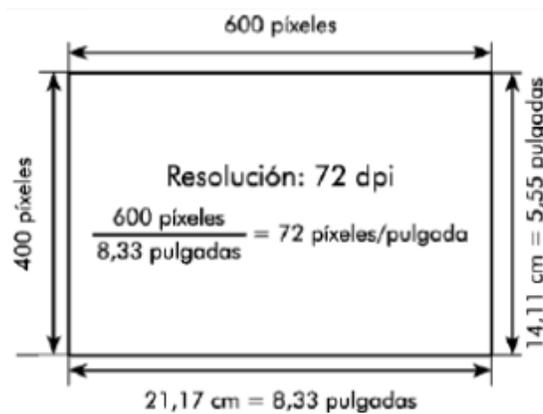
La profundidad de píxel es una unidad de medida binaria porque cada píxel está formado por bits. Cuando decimos que la profundidad de píxel es de 1 bit, la imagen solamente tiene dos colores o dos niveles de gris representados por los valores 0 y 1; por ejemplo solamente puede ser o blanco o negro.

### Tamaño de la Imagen

Se define con las dimensiones en pixeles de la matriz o cuadrícula

### Resolución

Es la medida de cantidad de pixeles por unidad de longitud, usualmente se utiliza los pixeles por pulgada que se abrevia “ppp”, o “dpi”. Como la resolución mide el número de pixeles por unidad de longitud, se deduce que a mayor resolución, mayor número de puntos de imagen, en el mismo espacio y, por tanto, mayor definición.



## Objetivos del procesamiento digital de Imágenes

En general, el procesamiento digital de imágenes busca resolver dos objetivos:

- Mejoramiento. Se busca incrementar alguna característica de la imagen.
- Restauración. Se busca eliminar perturbaciones en la imagen.

El mejoramiento no añade información adicional, sólo acentúa ciertas características para posterior análisis; ejemplo de estos procesos es el coloramiento selectivo, incremento de bordes, normalización, etc.

Una vez habiendo reforzado alguna característica, es posible realizar operaciones de análisis tales como búsqueda de patrones, detección de fallas, etc.

La restauración remueve degradaciones ya provenientes del ruido en el medio, ya provenientes de desperfectos en los instrumentos de captura, es decir, cuando la imagen digital no es generada mediante un dispositivo digital, sino mediante un proceso de conversión analógico-digital.

El ruido es de naturaleza completamente aleatoria, por lo que puede clasificarse por la distribución de probabilidad que lo modela así como por su distribución de potencia. El proceso de ruido más estudiado corresponde al ruido blanco Gaussiano: es ruido que tiene todas las componentes frecuenciales y que se puede extraer de una distribución Gaussiana.

Para restaurar una imagen digital se recurren a dos clases de técnicas: estadísticas y descriptivas.

Métodos estadísticos: mediante técnicas como Riesgo mínimo de Bayes, se puede restaurar una imagen capturada por un sistema no lineal.

Métodos descriptivos: se basan en la teoría de regularización de Tikhonov, en la cual se minimiza un funcional objetivo, compuesto por un término de fidelidad de datos y otro de regularización. Se supone conocimiento previo del modelo de adquisición de datos y de la estadística del ruido.

## Desarrollo de la práctica

1. Lea una imagen a color en cualquier formato y almacénela en la variable "Im".
2. Convierta esa imagen a una imagen en escala de gris (escriba su propio algoritmo).
3. Calcule el histograma de la imagen en escala de gris obtenida anteriormente (escriba su propio algoritmo).
4. Manipule la imagen para incrementar o decrementar el brillo de la imagen en escala de grises, según su criterio pero basándose en el histograma de la imagen. En otras palabras si del análisis del histograma determina que la imagen es muy oscura, incremente el brillo de la misma, si por el contrario determina que es una imagen con sobreexposición (muy blanca), reduzca el brillo de la misma.

## Reportar

- Código e imágenes procesadas
- Observaciones y conclusiones

# Guía de procesamiento de imágenes mediante MATLAB

## Lectura de Imágenes

Mediante el comando:

```
Imagen=imread('nombre.extensión')
```

```
Imagen=imread('nombre','formato')
```

La imagen a leer debe encontrarse en la carpeta de trabajo de Matlab. Los formatos de imagen soportados por Matlab son:

<b>Formato</b>	<b>Extensión</b>
TIFF	.tiff
JPEG	.jpg
GIF	.gif
BMP	.bmp
PNG	.png
XWD	.xwd

En Matlab una imagen en formato de color RGB se representa por tres matrices bidimensionales, correspondientes a los planos R, G y B.

Para obtener los planos R, G y B se ejecutan los comandos:

```
Im_R=Imagen(:,:,1)
```

```
Im_G=Imagen(:,:,2)
```

```
Im_B=Imagen(:,:,3)
```

```
[m,n,p]=size(Im_RGB)
```

m = 600

n = 800

p = 3

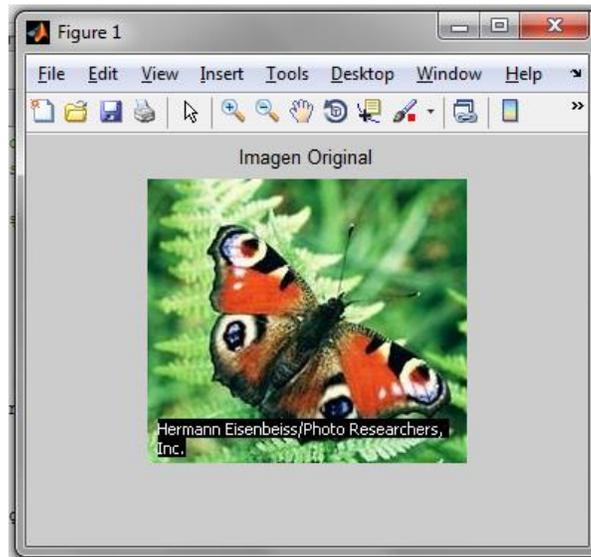
Im\_RGB: 600x800 3 planos (R,G y B)

## Despliegue de Imágenes

Se realiza con el comando:

```
imshow(Imagen)
```

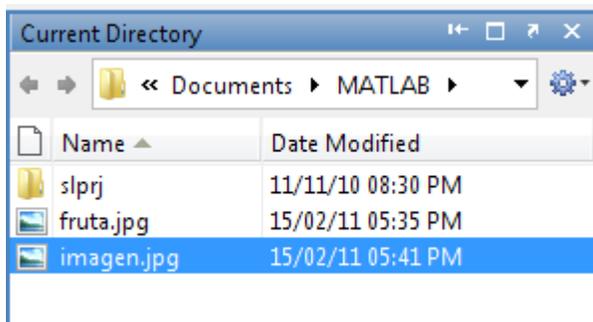
Dónde: Imagen es del tipo uint8.



## Escritura de Imágenes

Con el comando:

```
imwrite(Imagen,'nombre.extensión')
```



## Lectura del Valor de un Pixel

Obtención de valor de pixel

Imagen(m,n) ; Cuando Imagen está en escala de grises (un solo plano).

Imagen(m,n,p) ; Para imagen RGB. Devuelve el valor del pixel correspondiente al plano p (1, 2 ó 3).

Dónde: m,n son las coordenadas del pixel.

### Edición de Pixeles

Para modificar el valor de un pixel:

Imagen(m,n)=x ; Para una imagen en escala de grises.

imagen(m,n,p)=x ; Para una imagen RGB.

Dónde: x es un número entero entre 0 y 255 correspondiente a escala de grises (0=negro y 255=Blanco)

```
>> Im_RGB(200,750,1)=255;
```

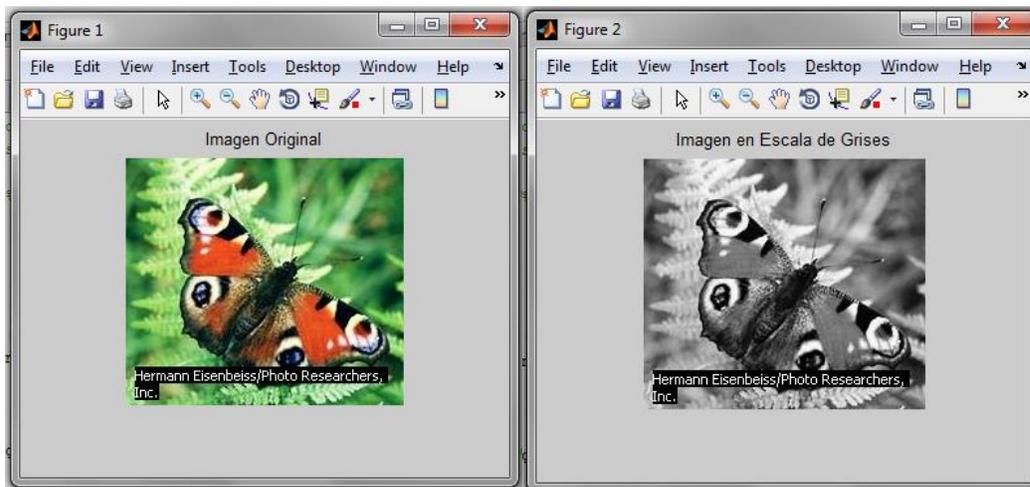
```
>> Im_RGB(200,750,2)=255;
```

```
>> Im_RGB(200,750,3)=255;
```



### Conversión a Niveles de Gris

Para convertir una imagen en color a niveles de gris

$$Im_{gris} = rgb2gray(Im\_RGB);$$


### Submuestreo

Submuestrear una imagen reduce su tamaño y permite que el procesamiento posterior de la imagen se agilice.



Se toman píxeles equidistantes (muestras), dependiendo del factor elegido, y se desecha el resto de píxeles.

$$Imagen\_sub = Imagen(1:a:end, 1:a:end, 1:1:end)$$

Dónde: a es el factor de muestreo.

Si a=2 la imagen se reduce a la mitad.

### Sobremuestreo

Si se desea incrementar el tamaño de una imagen, el programa debe crear nuevos píxeles para rellenar los huecos que aparecen al ampliar, dado que donde antes había 1 píxel ahora tendremos varios dependiendo del porcentaje de aumento de resolución. Esos nuevos píxeles se crean por interpolación, basándose en la información sobre el color de los píxeles vecinos.

```
Im_Interpolada = imresize(Im_RGB, factor,'metodo');
```

Donde Método: 'nearest', 'bilinear', 'bicubic'

### Códigos de MATLAB:

1.

```
close all hidden;
clear;
clc;
Normalize = @(x) (x - min(x(:)))/(max(x(:)) - min(x(:)));

N = 512;
N0 = 0.005;
K = 12;
Alpha=35;
b=zeros(N,1);
for i=1:floor(K/2),
b(i+1)=abs(sin(i/floor(K/2))/(i/floor(K/2)));
end
b(1)=1;
S=toeplitz(b);
im= double(rgb2gray(imread('Artificial.jpg')));
u = S*im;
u = u/max(max(u));
Noise=randn(N)*sqrt(N0);
u = u+Noise;
imtool(u);
est=inv(S'*S+Alpha*eye(N))*S'*u;
est = Normalize(est);
imtool(est);
```

2.

```
clear;
Normalize = @(x) (x - min(x(:)))/(max(x(:)) - min(x(:)));
im = zeros(512);

im(256-(128/2):256+(128/2),256-(128/2):256+(128/2)) = 1;
imtool(im);
imx = im(2:512,:);
imy = im(:,2:512);
```

```
im2(1:511,:)=abs(imx-im(1:511,:));  
im3(:,1:511)=abs(imy-im(:,1:511));  
im2 = [im2;zeros(1,512)];  
im3 = [im3 zeros(512,1)];  
imf = im2+im3;
```

```
imf = Normalize(imf);  
imtool(imf)
```