

UNIVERSIDAD POLITÉCNICA DE MADRID



*DEPARTAMENTO DE INGENIERÍA
ELÉCTRICA, ELECTRÓNICA,
AUTOMÁTICA Y FÍSICA APLICADA*

Prácticas de Visión Artificial

Práctica 1

***Iniciación al procesamiento de imágenes
con Matlab***

1		3
1.1	PRIMERAS OPERACIONES.....	3
1.2	GENERANDO UN FICHERO *.M	5
1.3	CONSTRUCCIÓN DE IMÁGENES.....	5
1.4	EJERCICIO.....	5

Iniciación al procesamiento de imágenes con Matlab

1

En esta práctica se iniciará el tratamiento de imágenes digitales con Matlab. **MATLAB** (abreviatura de *MATrix LABoratory*, "laboratorio de matrices") es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas de Unix, Windows y Apple. Entre sus prestaciones básicas se encuentran: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes. Además, se pueden ampliar las capacidades de MATLAB con las *cajas de herramientas (toolboxes)*. Para el caso de manipulación de imágenes se emplea el *toolbox "Image Processing"*.

1.1 Primeras operaciones

Una vez arrancado Matlab, el comando **ver** indicará los toolboxes instalados y su versión correspondiente:

```
ver
```

El primer comando que se empleará será la lectura de ficheros de imágenes (**imread()**):

```
imgEnt = imread('cameraman.tif');
```

imgEnt representa el identificador a la imagen leída. Para cualquier ayuda sobre las funciones o comandos se empleará el comando **help**:

```
help imread
```

Más información sobre las variables es obtenida observando la ventana **WORKSPACE**. También puede usarse el comando **whos**:

```
whos
```

Se puede comprobar que la imagen es de niveles de grises con 256 valores

diferentes (de la clase **uint8**), con un tamaño de 256 x256 píxeles. La visualización de la imagen es realizada usando **imshow**:

```
imshow(imgEnt);
```

Repita estas operaciones con la imagen '**mri.tif**'. ¿Cuál es su tamaño? ¿Cuántos niveles de grises tiene?. Si la imagen fuese en color, normalmente quedará definida por tres matrices correspondiente a los tres colores básicos (rojo, verde y azul). Vuelva a realizar las mismas operaciones de: a) lectura, b) tamaño y clase de la imagen y c) visualización sobre la imagen de color '**board.tif**'.

Usando la notación de matrices de Matlab se puede visualizar las tres componentes de color. El operador **:** hace referencia a todos los elementos de esa dimensión, luego el nivel de gris para cada parte del espectro de la luz será definido por **(:,:,i)**. Indica que todas las filas y las columnas para la componente i, i=1,2 ó 3 (rojo,verde,azul):

```
imshow([imgEnt(:,:,1),imgEnt(:,:,2),imgEnt(:,:,3)])
```

El operador **[]** permitirá construir una matriz de N x (3*M), siendo N el número de filas y M el número de columnas.

Emplee el comando **imtool** para ver el nivel de gris de la imagen de '**cameraman.tif**' y los colores en '**board.tif**'. Utilice el inspector de valores de los píxeles:

```
imtool('cameraman.tif');
```

```
imtool('board.tif');
```

Otro tipo interesante de imágenes son las binarias. Normalmente se emplea el '0' para indicar el fondo y '1' el objeto. Se empleará una técnica de *umbralización* para convertir las imágenes en binarias (**im2bw()**):

```
imgEntGris = imread('rice.png');
```

```
figure(1); imshow(imgEntGris);
```

```
imgBW = im2bw(imgEntGris);
```

```
figure(2); imshow(imgBW);
```

```
impixelinfo;
```

Realice la misma operación de binarizado con la imagen '**coins.png**'.

1.2 Generando un fichero *.m

En este apartado se tratará de realizar la primera función (*.m) de procesamiento de imágenes con Matlab. Consistirá en leer un fichero de imagen 2D, cuyo nombre es pasado por parámetro, se visualizará y se aplicará una umbralización automática, la cual es también visualizada.

```
function miPrimerScript(nombreFich)
% Lectura del fichero
imgEnt = imread(nombreFich);

%Visualización
clf;
figure(1);
imshow(imgEnt);

%Umbralización
imgBW = im2bw(imgEnt);
figure(2);
imshow(imgBW);

end
```

1.3 Construcción de imágenes

Se desea construir una imagen binaria de 120 x 200 píxeles que tenga franjas horizontales de 20 píxeles de anchura, distanciada por cada 20 píxeles:

```
imgBW = false([120,200]);

for i=1:40:100

    imgBW(i:i+19,:)=true;

end

imshow(imgBW);
```

Si se desea que las franjas sean verticales sólo habría que emplear el operador traspuesta de las matrices.

```
imshow(imgBW');
```

1.4 Ejercicio

Realizar una función que construya y visualice dos imágenes de 256x256 con variación del nivel de gris en filas y columnas.

