

Laboratorio 1: Fundamentos de Procesamiento Digital de Imágenes

Profesor Henry Arguello Fuentes, henarfu@uis.edu.co

7 de octubre de 2016

El informe del laboratorio debe entregarse el día 14 de octubre, en éste deben responder las preguntas que se presentan en la guía, además anexar los archivos que generen, por ejemplo .mat, .m, .fig, etc. Enviar una carpeta comprimida con los archivos al correo electrónico: jonathan.monsalve@saber.uis.edu.co, con el asunto **Laboratorio 1 imágenes** y el nombre de la carpeta comprimida **Lab1_NombreApellido**, donde Nombre es su primer nombre y Apellido es su primer apellido.

1. Objetivo

Comprender los principales conceptos del entorno MATLAB así como los comandos básicos IPT Toolbox de procesamiento de imágenes (Image Processing Toolbox por sus siglas en inglés).

2. Introducción

En MATLAB el toolbox encargado del procesamiento digital de imágenes (IPT) contiene diferentes comandos que permiten manipular imágenes.

Algunos de los comandos básicos más importantes se listan a continuación:

Comando	Descripción
<code>f=imread('nombre de archivo');</code>	Donde f es una matriz y G es el número de niveles de intensidad usados para mostrarla. Si G es omitido su valor por defecto es 256 niveles.
<code>imwrite(f,'nueva.jpg','compression')</code>	Las imágenes pueden ser escritas en disco usando la función <code>imwrite</code> .
<code>imshow(I)</code>	Muestra la imagen almacenada en I.
<code>B =logical(A)</code>	Una imagen binaria es arreglo lógico de 0s y 1s.

Existe una serie de imágenes que viene por defecto como parte de la caja de herramientas de procesamiento digital de imágenes. Para obtener una lista de las imágenes, se escribe: **help imdemos**

2.1. ¿Que hay en una imagen?

Una imagen es una matriz que puede ser accedida a través de índices. El formato general para indexar es (row,col) o (y,x), ¡Cuidado aquí porque se tiende a pensar en coordenadas cartesianas (x,y)!.

2.2. Mostrar información de una Imagen

La siguiente tabla resume algunas forma de obtener información a cerca de una imagen.

Comando	Descripción
<code>whos</code>	Obtiene información acerca del tamaño, el tipo y bytes, de todas las variables.
<code>whos I</code>	Para información a cerca de una imagen almacenada en I
<code>size(I)</code>	Obtiene el tamaño de la imagen almacenada en I
<code>class(I)</code>	Obtener información en I

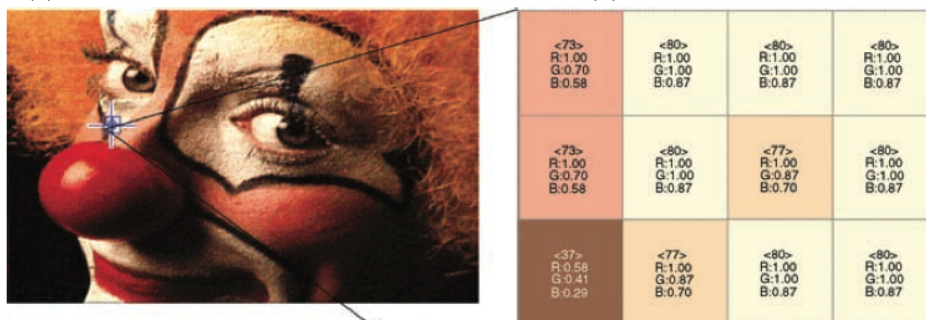
2.3. Formatos

Existen diferentes formas de almacenar imágenes. La siguiente tabla resume los diferentes tipos de imágenes y valores.



(a) Imagen binaria .

(b) Imagen a escala de grises.



(c) Imagen a color.

Para representar imágenes de verdadero color se debe agregar otra dimensión. Por ejemplo, para hacer rojo puro, rojo debe ser 255, verde debe ser 0, y azul debe ser 0.

Comando de MATLAB	Operación
<code>gray2ind()</code>	Convierte de formato de intensidad a formato indexado.
<code>ind2gray()</code>	Convierte de formato indexado a formato de intensidad.
<code>ind2rgb()</code>	Convierte de formato de indexes a formato de RGB
<code>mat2gray()</code>	Convierte de una matriz regular a formato de intensidad por escalamiento.
<code>rgb2gray()</code>	Convierte de formato RGB a formato de intensidad
<code>rgb2ind()</code>	Convierte de formato RGB a formato de indice
<code>im2bw()</code>	Convierte de intensidad/indices/RGB imagen a un umbral de negro y blanco en imagenes binarias.

3. Experimentos

- 1 Cargue la imagen coins.png, ejecutando el siguiente comando:

```
I = imread('coins.png');
```

Encontrar: Tamaño en bytes. Número de píxeles. Número de bits por píxel.

Ahora: Ejecute el siguiente script y compare los tamaños de archivos

```
imwrite(I, 'coins.png');  
imwrite(I, 'coins.jpg');  
imwrite(I, 'coins5.jpg', 'quality', 5);
```

- 2 Cargue la imagen 'parrots.tif' y examine las características de la imagen con el comando **whos**. *Nota: leer acerca del comando **colormap***

```
[I, map] = imread('parrots.tif');
```

- Almacene cada canal de color en diferentes variables y grafíquelas con el comando **imagesc** y defina un **colormap** para cada canal.
 - Genere un colormap de N entradas mediante la aproximación del colormap de la imagen original almacenado en la variable **map**. Grafique la imagen con **imagesc** y el nuevo colormap. *¿Cuántos colores son necesarios para ver una buena imagen?*
 - Convertir la imagen de **uint8** RGB a escala de grises y grafique con el comando **imagesc** y **colormap(gray)**.
 - Ahora cambiar el rango [0 255] a un rango mas bajo [0 N]. Grafique y examine el resultado para diferentes valores de N con el comando **whos**. *¿Hasta que valor de N no existe distorsión en la imagen?*
- 3 Realizar una función en matlab que contruya y visualice dos imágenes de 256x256 con variación de N niveles de grises en filas y columnas.
- 4 Escribir una función en Matlab que cree una figura sobre una matriz binaria. La figura puede ser un cuadrado, un rectangulo, un triangulo o un circulo. La función debe cumplir el siguiente formato:

```
function [I] = figura(stringfigura)  
% Genera una imagen binaria con la forma 'stringfigura';  
% -----  
% stringfigura = 'cuadrado', 'circulo', 'triangulo', 'rectangulo' o 'todos';  
end
```

La salida es de la forma:

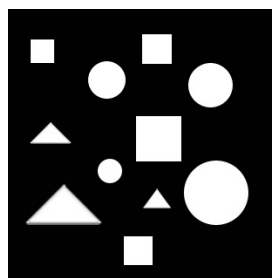


Figura 2: Matriz binaria

- 5 Usando la funcion creada en el punto anterior, genere las mismas figuras pero sin relleno (solo contorno).
- 6 El centro de masa es la posición geométrica de un cuerpo rígido, en la cual se puede considerar concentrada toda su masa (punto de equilibrio). Para este ejercicio calcule el área y el centro de una figura creada con el función figure.