

Práctica 0

Manejo Básico de Imágenes

Gutiérrez Castillo Óscar, Joya Venegas Jehosua Alan, Nolasco Fuentes Juan Carlos

Resumen – En esta práctica veremos distintas formas de desplegar una imagen en distintos formatos, así como algunas manipulaciones básicas para el procesamiento de imágenes.

I. INTRODUCCION

MATLAB

Instrumento Control Toolbox™ permite conectar MATLAB® directamente con instrumentos como osciloscopios, generadores de funciones, analizadores de señales, fuentes de alimentación e instrumentos analíticos. Esta toolbox conecta con sus instrumentos a través de controladores de instrumentos como IVI y VXIplug&play, o bien mediante comandos SCPI de texto a través de protocolos de comunicación de uso habitual como GPIB, VISA, TCP/IP y UDP. También incluye un conjunto de funciones (algoritmos) y herramientas gráficas enfocadas al procesamiento, análisis y visualización de imágenes, este conjunto es denominado toolbox de imágenes.

- imágenes Indexadas:

Consisten de una matriz de datos y una paleta de colores. La paleta de colores es un arreglo de m filas por 3 columnas la cual contiene valores de punto flotante en el rango de $[0,1]$. Cada fila de la paleta especifica las componentes roja, verde y azul para un color único. El color de cada pixel es determinado usando el correspondiente valor en la matriz de datos como un índice dentro de la paleta asignada. El valor 1 apunta a la primera fila de la paleta, el valor 2 a la segunda fila, etc.

- Imágenes de intensidad:

Consisten de una matriz de datos, cuyos valores representan intensidades dentro de algún rango, cada elemento en la matriz corresponde a un pixel en la imagen representando niveles de gris, donde la intensidad 0 usualmente representa al negro y la intensidad 1, 255 o 65536 usualmente representa total intensidad, o blanco.

- Imágenes RGB o a color:

Una imagen RGB, a veces es referida como una imagen de color verdadero, en MatLab® es almacenada como una matriz de $m \times n \times 3$ (m filas por n columnas y una profundidad de 3) que define las componente roja, verde y azul para cada pixel

individual. Las imágenes RGB no usan paleta de colores, el color de cada pixel es determinado por la combinación de las intensidades roja, verde y azul almacenadas en cada plano de color en la ubicación del pixel. Los formatos gráficos almacenan las imágenes RGB como imágenes de 24 bits, donde las componentes roja, verde y azul son cada una de 8 bits, esto da un potencial de 16 millones de colores. Un pixel cuyas componentes de color

FORMATOS DE IMÁGENES:

- JPEG: es el acrónimo inglés de Joint Photographic Experts Group, su formato es representado como .jpg o .jpeg. Es uno de los más populares debido a su habilidad para comprimir significativamente los datos con una pérdida visual mínima. Es un formato con pérdida, que comprime los datos usando la Transformada de Coseno Discreta (TCD). Este formato no es adecuado para almacenar imágenes que contienen estructuras finas como líneas, curvas etc.

- TIFF: es el acrónimo inglés de Tagged Image Format. Su extensión es .tif o .tiff, fue creado en la década de los 80's para almacenar y codificar documentos escaneados. Originalmente fue desarrollado para datos de un solo bit, pero hoy en día el estándar permite almacenar 16 bits incluso datos de punto flotante.

- DICOM: es el acrónimo inglés de Digital Imaging and Communication in Medicine. Es un formato estándar para codificar y transmitir Imágenes de Resonancia Magnética (MRI) e Imágenes de Tomografía Computarizada. Este formato almacena información junto con otros datos entre los que destacan; detalles del paciente, parámetros de adquisición, hora de creación, modelo del equipo, información del hospital etc. DICOM es usado en varias especialidades de medicina como Radiología, Neurología, Cirugía, Cardiología, Oncología entre otros.

II. PROCEDIMIENTO

En el folder de imágenes se le proporcionan otro tipo de imágenes que no son médicas, puede usted utilizarlas para el desarrollo de la práctica. Desarrolla un script para leer y desplegar cada imagen. De todas las imágenes que lea imprimir el tipo de imagen, el tamaño y el tipo de dato

EJERCICIOS

1. Abrir y visualizar la imagen RGB “magriclonRGB.jpg” y cada una de sus componentes de color por separado usando la paleta de colores roja, verde y azul.

Código:

```
rgbImage = imread('magriclonRGB.jpg','jpg'); % Read image
% Extract color channels.
redChannel = rgbImage(:,:,1); % Red channel
greenChannel = rgbImage(:,:,2); % Green channel
blueChannel = rgbImage(:,:,3); % Blue channel

% Create an all black channel.
allBlack = zeros(size(rgbImage, 1), size(rgbImage, 2), 'uint8');

% Create color versions of the individual color channels.
just_red = cat(3, redChannel, allBlack, allBlack);
just_green = cat(3, allBlack, greenChannel, allBlack);
just_blue = cat(3, allBlack, allBlack, blueChannel);

% Display them all.
subplot(3, 3, 2);
imshow(rgbImage);
fontSize = 20;
title('Imagen Original', 'FontSize', fontSize)

subplot(3, 3, 4);
imshow(just_red);
title('Canal rojo', 'FontSize', fontSize)

subplot(3, 3, 5);
imshow(just_green);
title('Canal verde', 'FontSize', fontSize)

subplot(3, 3, 6);
imshow(just_blue);
title('Canal azul', 'FontSize', fontSize)

% Set up figure properties:
% Enlarge figure to full screen.
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0, 1, 1]);
% Get rid of tool bar and pulldown menus that are along top of figure.
% set(gcf, 'Toolbar', 'none', 'Menu', 'none');
% Give a name to the title bar.
set(gcf, 'Name', 'Demo by ImageAnalyst', 'NumberTitle', 'Off')
```

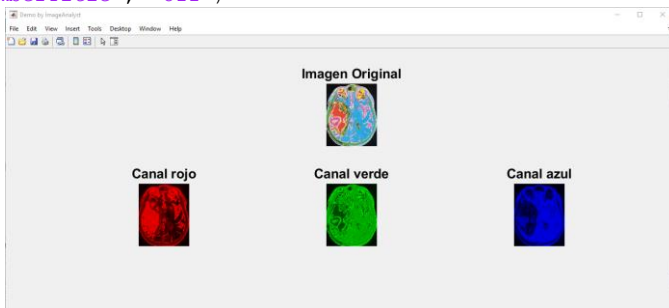


Fig. 1. Podemos ver en la imagen la imagen en diferentes colores por separado

2. Abrir y desplegar la imagen RGB “retinaRGB.jpg” y convertir cambiando el espacio de color de:

2.1. RGB a Escala de grises

2.2. RGB a YUV 2.3. RGB a HSV

Código

```
RGB = imread('retinaRGB.jpg','jpg');

escalaGrises = rgb2gray(RGB);

hsv = rgb2hsv(RGB);

subplot(2,3,2);
imshow(RGB);
fontSize=20;
title('Imagen Original','FontSize',fontSize)

subplot(2,3,4);
imshow(escalaGrises);
title('Escala de grises','FontSize',fontSize)

subplot(2,3,5);
imshow(hsv);
title('Escala HSV','FontSize',fontSize)

R=RGB(:,:,1);
G=RGB(:,:,2);
B=RGB(:,:,3);
Y = 0.299 * R + 0.587 * G + 0.114 * B;
U = -0.14713 * R - 0.28886 * G + 0.436 * B;
V = 0.615 * R - 0.51499 * G - 0.10001 * B;
YUV = cat(3,Y,U,V);
subplot(2,3,6);
imshow(YUV);
title('Escala YUV','FontSize',fontSize)
```

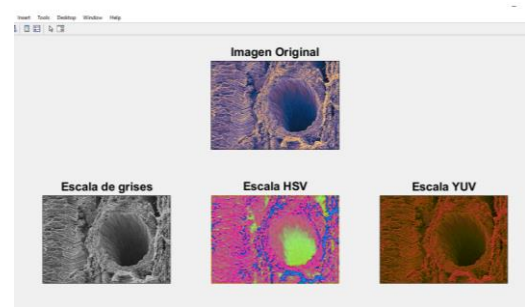


Fig. 2. Podemos ver en la imagen en escala de grises, HSV, YUV.

3. Leer y desplegar la imagen del archivo “corte.bmp”, y obtener una imagen cortada a un tercio de su tamaño, realice el corte (crop) en la esquina inferior derecha. Realice otra subimagen donde usted desee.

Código:

```
originalImage = imread("corte.bmp");
[rows, columns, numberOfColorChannels] = size(originalImage);

cropImage1 = imcrop(originalImage,[275,0,413,513]);
cropImage2 = imcrop(originalImage,[0,275,513,413]);
```

```
subplot(1,3,1);
imshow(originalImage);
```

```
subplot(1,3,2);
imshow(cropImage1);
```

```
subplot(1,3,3);
imshow(cropImage2);
```

```
subplot(1,3,2);
imshow(cropImage1);
```

```
subplot(1,3,3);
imshow(cropImage2);
```

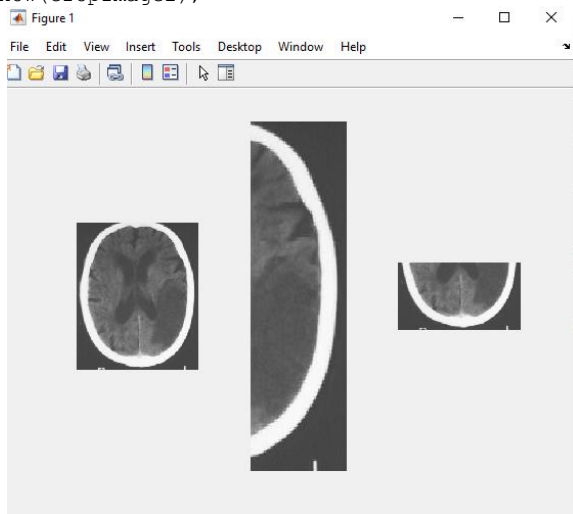


Fig. 3. Podemos ver en la imagen cortada en con diferentes proporciones.

4. Leer la imagen del archivo "rx cerv.pcx" y desplegarla usando primero el comando `image` y posteriormente el comando `imshow`.

Código:

```
rx cerv = imread('rx cerv.pcx','pcx' ); % Read image
%Showing the image with image
figure(1);
image(rx cerv);
```

```
%Showing the image with imshow
figure(2);
imshow(rx cerv);
```

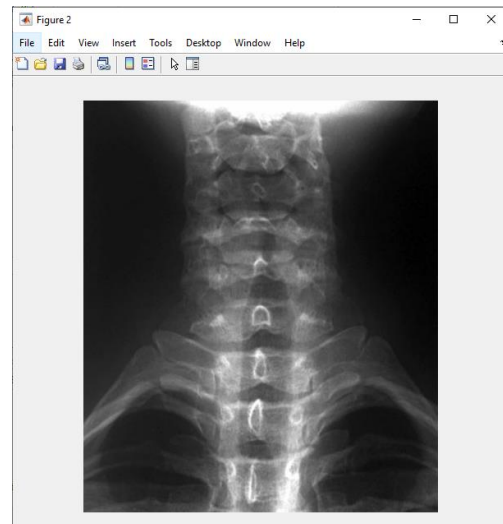


Fig. 4. Podemos ver en la imagen leída del archivo "rx cerv.pcx".

5. Leer y desplegar la imagen "abdomen.png" usando el comando `image` y usando posteriormente el comando `imagesc`.

Código:

```
subplot(1,2,1);
C = imread("abdomen.png");
image(C);
```

```
subplot(1,2,2);
C = imread("abdomen.png");
imagesc(C);
```

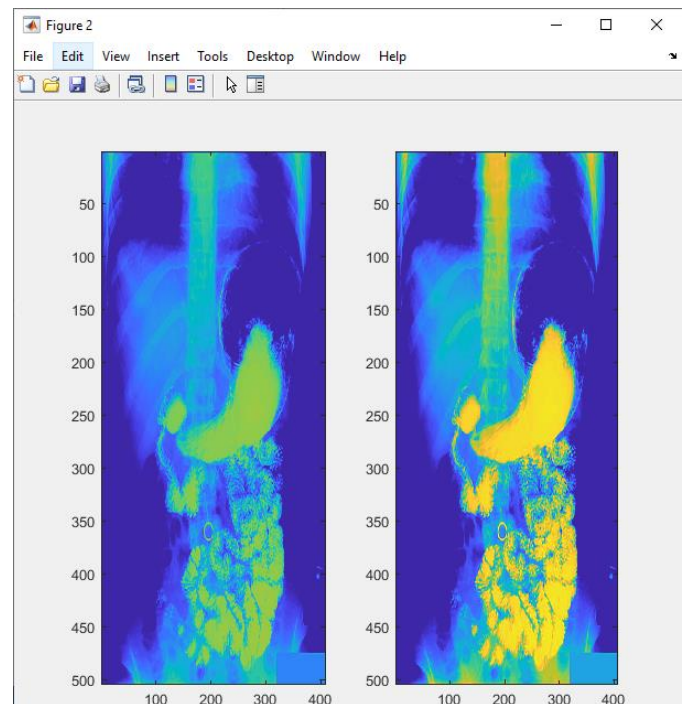


Fig. 5. Podemos ver en la imagen leída del archivo "abdomen.png" y aplicamos la función `imagesc`.

6. Despliega la paleta de colores de RGB por separado

Código:

```
magriclonRGB = imread('magriclonRGB.jpg','jpg'); %
Read image

% Extract color channels.
redChannel = rgbImage(:,:,1); % Red channel
greenChannel = rgbImage(:,:,2); % Green channel
blueChannel = rgbImage(:,:,3); % Blue channel

%Showing red colorbar
figure(1)
image(redChannel)
colorbar

%Showing green colorbar
figure(2)
image(greenChannel)
colorbar

%Showing blue colorbar
figure(3)
image(blueChannel)
colorbar
```

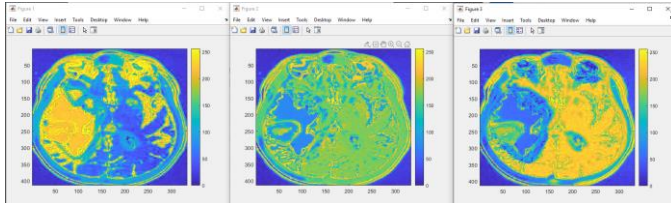


Fig. 6. Podemos ver en la imagen en colores separados por RGB.

7. Visualizar la imagen “intestinoRGB.jpg” con diferentes paletas de colores predefinidas de MatLab.

Código:

```
%Reading image
abdomen=imread("abdomen.png", 'png')

%Colormap jet
figure(1);
image(abdomen);
colormap("jet");

%Colormap autumn
figure(2);
image(abdomen);
colormap("autumn");

%Colormap bone
figure(3);
image(abdomen);
colormap("bone");

%Colormap copper
figure(4);
image(abdomen);
colormap("copper");

%Colormap winter
figure(5);
image(abdomen);
colormap("winter");
```

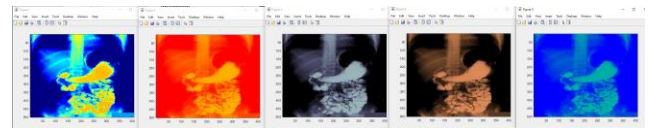


Fig. 7. Podemos ver en la imagen con las diferentes paletas de MatLab.

8. Visualizar la imagen indexada “cta_scan_index.bmp”. Rota la imagen a 45, 90 y 180 grados

Código:

```
cta_image = imread("cta_scan_index.bmp");
rotate45 = imrotate(cta_image,45);
rotate90 = imrotate(cta_image,90);
rotate180 = imrotate(cta_image,180);

subplot(2,2,1);
imshow(cta_image);

subplot(2,2,2);
imshow(rotate45);

subplot(2,2,3);
imshow(rotate90);

subplot(2,2,4);
imshow(rotate180);
```

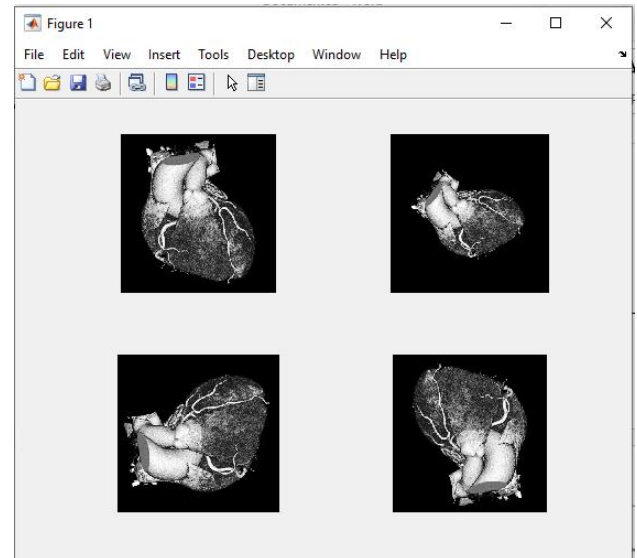


Fig. 8. Podemos ver en la imagen rotada.

9. Convierte la imagen mri.jpg a escala de grises, y guárdala en formato .tiff

Código:

```
RGB = imread('mri.jpg','jpg');

escalaGrises = rgb2gray(RGB);

imwrite(escalaGrises, 'mri.tiff');
```



Fig. 9. Podemos ver en la imagen que si genero el archivo tiff.

10. Imagen DICOM ó Imagen satelital:

Si usted decide usar DICOM: Realiza un script que lea las imágenes de la carpeta DICOM y muestra en una misma ventana las 280 imágenes de tal manera que se pueda simular el movimiento del corazón. Utiliza ciclos for y el comando pause, para leer la ruta completa de la carpeta.

SI usted decide usar una imagen satelital:

10.1. Lea la imagen satelital. Y enumere los componentes que tiene la imagen, así como la resolución que manejan. Despliegue dos de los componentes.

10.2. Reajuste el tamaño de la imagen disminuyendo 3 veces el tamaño original. Despliegue

```
for i = 1:280
    if (i < 10)
        imagen = dicomread("IM-0001-000"+i+".dcm");
    elseif (i > 9 && i < 99)
        imagen = dicomread("IM-0001-00"+i+".dcm");
    else
        imagen = dicomread("IM-0001-0"+i+".dcm");
    end
    imshow(imagen, 'DisplayRange', [])
    pause(0.01)
end
```

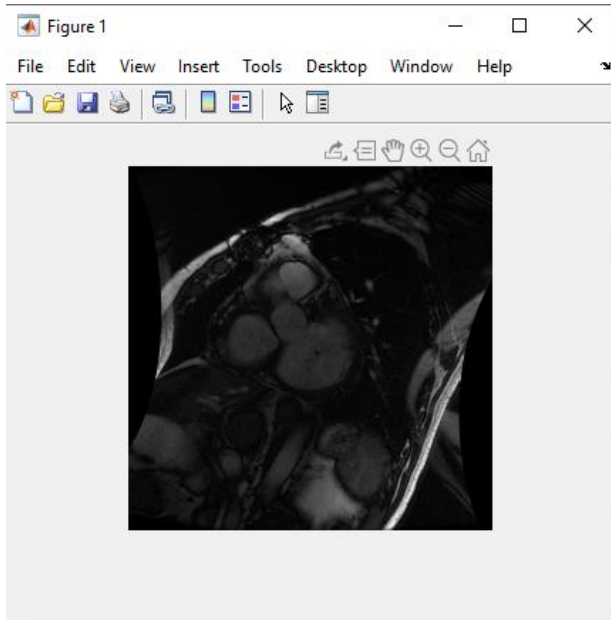


Fig. 10. Podemos ver en la imagen leída DICOM.

III. CONCLUSIÓN

En esta practica nos dimos una idea de lo robusto y amplia que es las herramientas que tiene Matlab. Estas herramientas nos proporcionan desde funciones para rotar hasta el tratamiento de imágenes. También en esta practica nos dimos ala tarea de investigar las funciones a las imágenes y nos encontramos con excelente documentación y ejemplificación de las funciones en la página oficial.

REFERENCES

- [1] MATLAB Documentation:
<http://www.mathworks.com/help/matlab/>
 [2] Digital Image Processing, González, R.C , Woods, P., Addison Wesley, 1992

[5] C. Solomon y T. Brekon, Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab, Oxford, UK: Wiley's global Scientific, Technical and Medical business with Blackwell Publishing., 2011.

[4] E. Moya Albor, Práctica 2: Manejo Básico de imágenes con Matlab, CDMX, 2015.