Images in MATLAB

The representation of images

A grayscale image is represented by a two-dimensional array. The first dimension is the row of a pixel and the second is the column. The content of a two-dimensional matrix **A** can be displayed using imshow(**A**). The type of **A** can be either **uint8** (values 0-255) or **double** (values 0-1).

A color image is represented by a three-dimensional array: row x column x color. The third dimension has a fixed size of 3 because three colors (red, green and blue) are used to describe the color of each pixel. The data type of the array can be **uint8** or **double**. For each color, 0 represents the darkest and 255 represents the brightest for uint8 (or 1.0 for double).

Displaying an image

```
> figure(1);
> imshow(A);
```

Alternatively, you can use the following:

```
> image( A );                 % displays the image in Figure 1
> axis image;                 % adjusts the axis such that pixels are
                              % equally spaced on the x and y axes.
> title( 'Original Image');   % prints a title on the image
```

To remove the numbers on the axes, use the following:

```
> axis off;
```

Reading an image from a file

```
>A = imread( 'test.jpg');
```

imread returns the image data in the array A. If the file contains a grayscale image, A is a two-dimensional (M-by-N) array. If the file contains a color image, A is a three-dimensional (M-by-N-by-3) array. The type of the returned array depends on the data type used by the file format but will be mostly uint8 for our purposes. Bmp, png, jpg, gif, tiff are supported. See MATLAB help for more supported types.

You can also directly display an image without reading it into a matrix.

```
> imshow('test.jpg');
```

Retrieving information about an image file

```
> A = imread( ' test.jpg');
> dim = size( A );
> y_dim = dim( 1 );            % assign the height of the image to variable y_dim
> x_dim = dim( 2 );            % assign the width of the image to variable x_dim

        OR

> info = imfinfo( 'test.jpg');
> y_dim = info.Height          % assign the height of the image to variable y_dim
> x_dim = info.Width           % assign the width of the image to variable x_dim
( see imfinfo help for all the returned properties)
```

Writing an image

```
> imwrite(A, 'test_out.jpg', fmt);      % format can be 'bmp', 'gif', 'jpg' etc.
```

Creating synthetic images

An example using the **uint8** type:

```
function red_sq_green_sq
% This function creates a 100x500 pixel blank image
% and then draws two squares and a line between them

    y_dim = 100;  % rows
    x_dim = 500;  % columns

    x = uint8( zeros( y_dim, x_dim, 3 ) );% create array with y_dim
                                          % rows x_dim cols and 3 bytes
                                          % for each pixel
     % draw red square
    x(25:75,100:150,1) = 255;

     % draw green square
    x(25:75,350:400,2) = 255;

     % draw a blue line between them
    x(48:52,151:349,3) = 255;

    imshow(x); % assumes a range of 0 - 255 when uint8
```

Another example using the **double** type:

```
function yellow_sq_purple_sq
% This function creates a 100x500 pixel blank image
% and then draws two squares and a line between them

    x_dim = 500;  % columns
    y_dim = 100;  % rows

    x = zeros( y_dim, x_dim, 3 );% create array with y_dim
                                 % rows x_dim cols and 3 bytes
                                 % for each pixel
     % draw yellow square
    x(25:75,100:150,1:2) = 1;

     % draw purple square
    x(25:75,350:400,[1 3]) = 1;

    % draw a gray line between them
    x(48:52,151:349,:) = .7;

    imshow(x); % assumes a range of 0-1 when double
```