

CMSC 426, Image Processing

Short Tutorial on MATLAB

Prof. Yiannis Aloimonos,
Nitin J. Sanket and Kiran Yakkala

August 31, 2016

Pin It!

You are given an image of colored objects a white background. Your task is to segment out the objects, count the number of colored objects and also count the objects of the same color, i.e., green, blue, yellow and red. To have some fun, we also threw in a white object and a transparent object (we really want you to try to get these as well to make your fundamentals stronger).

These are the functions which might help you in detecting these objects.

- **imread**: This is used to read an image from a file, use the command `help imread` or `doc imread` on the command window for more details on the function usage.
- **imshow**: Displays the image (Note: Any 2D/3D array can be displayed using this function, fun to ‘see’ how a matrix ‘looks’).
- **im2double**: Convert image to double. Be careful to see what data-type you are using, some functions only work with `uint8` and some with `double`. I would recommend using `double` for all your operations. Note that `uint8` acts as a double ended clipper clipping your values between 0 and 255.
- **rgb2gray**: Convert RGB image to grayscale.
- **im2bw**: Convert an image to black and white (BW).
- **fspecial**: Generates a 2D filter to be applied on the image.
- **imfilter**: Used in conjunction with `fspecial` generally, used to apply any filter onto an image.
- **imhist**: Compute and display image histogram.
- **imrotate**: Rotate an image.
- **imresize**: Resize an image (Make it bigger or smaller).

When you are working with RGB images (say image I), the array/matrix is of size $M \times N \times 3$ in the order Red, Green and Blue. These can be accessed as $I(:, :, 1)$ for Red channel, $I(:, :, 2)$ for Green channel and $I(:, :, 3)$ for Blue channel. This will be a useful tool to have in your arsenal when trying to segment out different colors (Think about what makes a red object different from a blue object and so on).

You can condition a matrix(I) with conditions to obtain a binary mask. For example say that the color values are bounded between 0 and 1 and you want to find a mask which denotes when the value is greater than 0.5. This is done using $I > 0.5$.

After you have obtained a mask you might want to remove some blobs (group of connected white pixels) or do some further processing on them (will come in handy in counting number of objects or rejecting stray stuff). A class of functions which help you in analysing the shape and structure of blobs in a BW image are called “Morphological Operations”. Some of the MATLAB’s built-in Morphological Functions are given next.

- **imdilate**: Makes the blobs bigger in a BW image.
- **imerode**: Makes the blobs smaller in a BW image.
- **bwmorph**: This has a set of BW morphological operations. This is pretty useful for ‘cleaning’ BW images.

You might find the function **bwlabel** helpful to label/extract individual blobs in the image. Helps in counting the blobs. The function **regionprops** can be used to extract various properties of different blobs. Some tools which will help you in debugging of computer vision algorithms are given next.

- **plot**: Plots the values on the axis.
- **stem**: Generally used for discrete plots.
- **hold**: **hold on** retains plots in the current axes so that new plots added to the axes do not delete existing plots. **hold off** sets the hold state to off so that new plots added to the axes clear existing plots and reset all axes properties.
- **imtool**: Opens image viewer app. Note windows opened by **imtool** are not closed by using the command **close all**.

If you want to change the dimensions of a matrix use the command **reshape**. To remove singleton dimensions, i.e., dimensions whose size is 1, use the command **squeeze**.