Image smoothing and simple geometric operations in Matlab

September 2016

Abstract

In this session, we are going to work with the smoothing image processing toolbox in MATLAB. In the folder downloaded from the Campus Virtual you will find all the material you will need for this practicum. Follow the instructions and answer the questions.

Contents

1 Creating images of 3 channels (color images)

The RGB images are formed by 3 matrices, commonly called channels.

Create a file "exercise1.m" to implement the following steps:

1. Create the 3 images in gray scale as shown in Figure 1.
2. Combine the 3 obtained images to construct the color image shown in Figure 1(right).
3. Save the color image as 3channels.jpg.

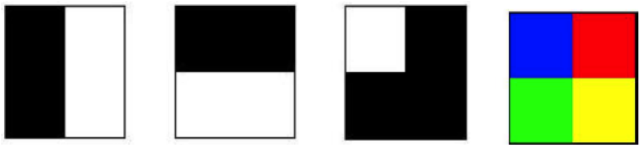

Figure 1: Displaying color images.

2 Displaying color images

Create a file "exercise2.m" to implement the following steps:

1. Read the image chairs.jpg.
2. Display different channels and explain the difference and similarities in pixel values.
3. What would happen if we interchange the channels?
4. What would happen if we multiply one of them by 0?

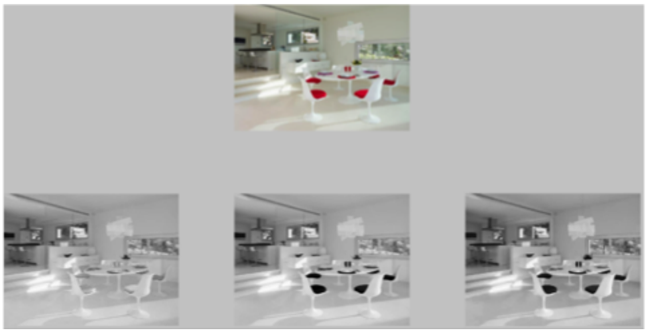

Figure 2: Chairs image for displaying color images.

3 Managing different size and filters

Create a file "exercise3.m" to implement the following steps:

1. Read one of the images in the folder images.
2. Show how details of the image disappear when rescale (make smaller) the size of the image (Help: imresize).
3. Does the histogram change of both images (the original and the rescaled one)?
4. Return back the smaller image to the original size. Compare to the original one and make a comment on that.
5. As an alternative of removing image details, apply different smoothing filters (user-defined mask as a vector e.g. [1 1 1] vs. Gaussian filter). Discuss how the size of the filter or mask affect the final outcome.
6. Can you apply the filter on the rgb image? What type should be the image before convolving and why? What dimensions should be the mask?
7. What is the difference using the following three masks: [1 1 1 1 1], [1;1;1;1;1] and [[1 1 1 1 1]; [1 1 1 1 1]; [1 1 1 1 1]; [1 1 1 1 1]; [1 1 1 1 1]]? Do we need to normalize the mask (divide by the sum of all its numbers)? Apply several times in order to observe the effects.
8. You can subtract the original and smoothed images in order to illustrate the difference between them. Use subplot in order to show the original smooth and the difference image in the same figure.

Note: Repeat these experiments with a couple of other images. If you have implemented all exercises as functions it should be immediate.

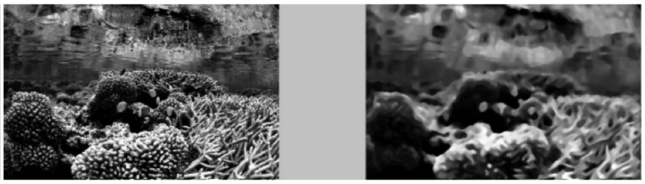

Figure 3: Managing different size and filters.

4 Simple geometric operations

Follow the instructions in order to create the images in Figure 4: Read the image clooney.jpg and change the place of both figures so that George Clooney stands on the left. The function should have 2 parameters: the original image and a number being the column number of the cut. It should return a new image with the persons interchanged. (You can use imcrop to find the optimal cut column.)

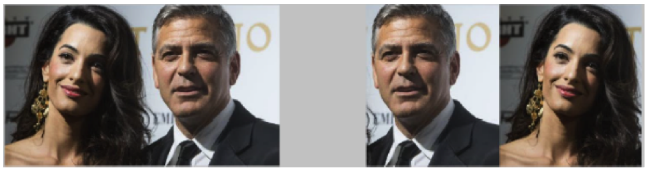

Figure 4: Simple geometric operations.

5 Image binarization

The binarization $B_I(x, y)$ of an image $I(x, y)$ from a threshold (Th) consists in turning the image into a binary image (of 0s and 1s) that will depend on whether the level of the pixel intensity of the original image is above or below a threshold.

$$B_I(x, y) = \begin{cases} 0, & \forall(x, y) : I(x, y) < Th \\ 1, & \forall(x, y) : I(x, y) \geq Th \end{cases}$$

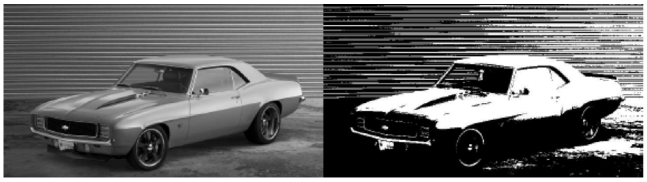

Figure 5: Thresholding: right image represents the binarization of left image using Th=128.

Given the image car_gray.jpg, create the function thresholdImg() which implements the following points:

1. Create the binary version of the original image by a threshold value of 20. What does it happen if we use different threshold values (30, 150, 255)? Why?
2. Visualize and save the image of threshold 150 as car_binary.jpg.
3. What will happen if you multiply the original image by the binary image?
4. What will happen if you multiply the original image with the inverted binary image? Help: im2bw(), imtool()

6 Treating color images

Given the images hand.jpg (Figure 3(left)) and mapfre.jpg (Figure 3(middle)), create the function fuseImg(), which implements the following points:

1. Open hand.jpg and convert it in gray scale image.
2. Perform a binarization to obtain a binary image of 2 regions: the hand (called foreground) and the rest (called background). Create the inverse binary image changing the areas of foreground and background.
3. Use the binary matrices created in (2) to merge the images hand and mapfre (Fig. 3(right))
4. Save the image as hand_mapfre.jpg.

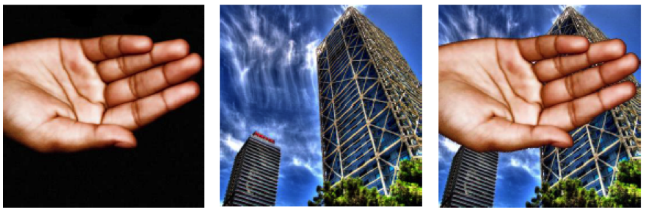

Figure 6: Overlapping images.

7 Practicum submission

The evaluation of the practicum will be based on the code and report that must be submitted (via campusvirtual2.ub.edu) in a file "StudentName1+StudentName2_CV_Lab1.zip". More specifically, the submission should contain:

- A report entitled "Introduction to Matlab" including the results of the problems properly commented and all necessary images to fully understand your discussion. The report must provide answers for all questions, results obtained and conclusions about them, as well as observations and difficulties found during the procedure.
- The files with the solution of the exercises. The Matlab code should be properly commented, including description of the developed functions.

Deadline: 10 of October, 23:55h by Campus Virtual.