Multimédia e Novos Serviços
(EIC0064) 2020-2021

Second Lab Assignment - Lab 2: Principles of Visual Processing in Matlab

Objectives: This assignment intends to apply and consolidate the knowledge gained concerning visual media representation in the digital domain and; to get acquainted with some processing techniques to improve images.

Introduction

The objective of this laboratory assignment is to introduce experimentally the student to different representations of digital visual signals and to some basic visual processing techniques, by using Matlab.

To conduct the proposed experiments, Matlab scripts and images available on Moodle will be used. Images will be use as input data to those scripts. The output will consist of processed versions of those images, which should be analysed by the student to interpret the effects of the conducted processing operations. The majority of those images has the bitmap format (.bmp) which means that each pixel is represented by three RGB eight-bit values, so in total 24 bits per pixel. Comparing results obtained by using different algorithms it is intended that the student acquires a better understanding of the role played by the different techniques and formats.

Note: the symbol ✍ means that you should include in your report graphics or images resulting from the operated processing or code that you may have developed. The symbol 🕵 indicates that you should include in your report a brief analysis of the results you have obtained.

Work to be developed

1. *Color spaces*

There are several color spaces to represent images, each one with their own coordinate system and having different areas of application or purposes. In this part of the assignment you will be using three different color spaces: RGB (Red, Green, Blue), HSV (Hue, Saturation, Value, where V represents brightness) and YUV (Luminance and two color-difference signals.

1.1. Based on the set of built-in functions presented in "introduction to images in Matlab" , write a um script Matlab ✍ that:

i) imports a bitmap image and presents that image on the screen;

ii) separate each RGB component in a different matrix and visualise each one on the screen ✍;

iii) convert the RGB image to the HSV color space and present it on the screen;

iv) separate each HSV component in a different matrix and visualise each one on the screen ✍;

Run the script with the images "testRGB.bmp", "floresVermelhas.bmp", "folhasVerdes.bmp", "praia.bmp" and "elephant.bmp". Compare the components of each image 🕵. You may experiment with other bitmap images (.bmp).

1.2 Write a similar script but that converts from RGB to para YCbCr.

Run the script with the images "testRGB.bmp", "floresVermelhas.bmp", "folhasVerdes.bmp", "praia.bmp" and "elephant.bmp". Compare the components of each image 🕵. Compare with the results obtained with the previous script.

1.3 Use the script "rgb2yuv.m" and verify if there are differences in relation to 1.2. 🕵

2. *Varying the spatial dimensions of an image with and without a filter, using the test image "imzoneplate"*

In this part you will use the Matlab scripts "ampliaReduz.m" and "imzoneplate.m" available on Moodle. Some of them are modified versions of programs available in the MathWorks Web site.

Before starting your work, analyse the code of the referred scripts so that you may have a full understanding of the operations that are performed.

The script that increases/decreases the spatial dimensions of an image, performs this processing using the test image "zone plate", which is generated during its execution by invoking the function "imzoneplate.m".

Start Matlab and change to your own working directory. Copy all the necessary files (scripts and images).

i) run the program "ampliaReduz.m" using different dimensions to the test image zoneplate and using different interpolation methods provided by the built-in function "imresize.m". Compare and interpret the results. 🕵

3. *Filtering experiments*

3.1. In these experiments you will use the program "filtro2019.m", which allows to perform different filtering operations to an image.

In these experiments you will use the program "filtro2019.m", which allows to perform different filtering operations to an image.

Open the program in the Matlab editor and verify which are the input parameters you should provide the script with. Run successively the script using different images and testing several possible filters. Verify and analyse the effects of each filter ✍.

For the cases of Mean and Gaussian filters, run the script using different values for the dimensions of the filter. compare and discuss the results. 🕵

3.2. In these experiments you will use the program "filtroMediana.m", which allows to eliminate punctual noise from images without removing sharpness to the image, through the application of a median filter.

Open the program in the editor and check the code to understand what the program does and what kind of information it operates on. Then run the program using different images (originally with and without noise) and testing different amounts of noise. Run the same images with average or Gaussian filters and compare and analyze the results. 🕵 ✍.

Note: there are available images with the desired type of random point noise for this experiment (‘ruido1.jpg’ and ‘ruido2.jpg’). You can use them to make the comparison. But you can also enter code in the “filterMediana.m” script to save the images in which the noise was introduced.

Report should be delivered no later than March 14 in Moodle.