Applications and New Services, MM0050
Multimedia Systems and Applications - M.EIC026
2023-2024

Lab Assignment 2

Assignment Name: introduction to visual signal processing

Due Date: March 22th, 2024. Submit in Moodle your report file in PDF and all the Matlab or notebooks/Python scripts with the code you have developed as a single compressed archive.

Purpose

This work intends to apply and consolidate knowledge about the representation of visual signals in the digital domain; to get acquainted with basic image processing techniques in Matlab and/or Python - color spaces; spatial sampling; and filtering; contour detection.

Why is it important that multimedia students gain the listed skills and knowledge? Because it provides them the foundations for the manipulation of visual signals. Because many multimedia systems perform such kind of operations to provide functionality to the user for modifying the signal appearance. Because in visual signals compression techniques, many times such type of operations are performed as preparatory steps.

Skills

In this assignment, you'll be learning how to:

1) develop simple practical algorithms using Matlab or Python to manipulate characteristics of visual signals (still images);
2) how to apply filters to still visual signal to extract some features, namely object counters, and
3) evaluate the efficiency of the selected techniques.

Knowledge

In this assignment, you'll be learning about:

1) different color spaces for representing visual signals;
2) the effects of spatial sub-sampling in visual signals;
3) the effects of applying different types of spatial filters to still images.

Task overview

The objective of this work is to allow students to acquire a better understanding of the fundamentals of representing visual signals in the digital domain and of processing techniques that allow for improving image quality, extracting low-level features and preparing signals for compression.
To carry out these experiments, whenever necessary, use images available on the UC's Moodle as input signals for the algorithms/processes implemented by the developed programs or scripts (some also available on the UC's Moodle page). Many of these images are in PNG (Portable Network Graphics) or BMP (bitmap) format, which means that each pixel is represented by three 8-bit RGB values, that is, a total of 24 bits. Comparing results obtained by applying different algorithms, it is intended that the student acquires a better understanding of the role played by different techniques and formats.

To familiarise yourself with Matlab commands and functions that manipulate images, please read the accompanying document “IntroMatlabVisual” available in Moodle. For Python Jupyter Notebooks some information is contained in the document “intro to Jupiter notebooks.pdf” available in the section Assignment 1.

Additionally, a number of example scripts for both Matlab and Python are provided in Moodle in the section devoted to Assignment 2.

Note: the symbol 📈 means that you must include in your report graphics or images that resulted from the processing carried out or even code that you have developed. The symbol ✍ indicates that you should include a brief analysis of the results you obtained in the report.

Implementation work to be carried out

1. Experiments with color spaces

There are several color spaces to represent visual signals, each with its own coordinate system, having different purposes or areas of application. In this part of the work we will use in total four color spaces: RGB (Red, Green, Blue), HSV (Hue, Saturation, Value, where V represents brightness), YCrCb/YUV (luminance and color difference signals), and L*a*b (L is the luminosity level, a is color channel along the green-red axis and b is the color channel along the blue-yellow axis. This color space is based in the “color opponent theory”, by which no color can be simultaneously green and red or blue and yellow).

1.1. Write a Matlab script that:

1.1.1. Imports an image in bitmap format (RGB color space) and display this image on the screen;

1.1.2. Separates each RGB component into a different matrix and display each one of them on the screen 📈;

1.1.3. Converts that image to HSV color space and display that image on the screen;

1.1.4. separates each HSV component into a different matrix and display each one of them on the screen 📈;

Run the script with various images such as “peppers.png”, “lighthouse.png” and others available in Matlab, or “floresVermelhas.bmp”, “folhasVerdes.bmp”, “praia.bmp” and “elephant.bmp”, available on Moodle. Compare components of each image with each other ✍.

1.2. Develop a similar script but instead of converting to HSV convert to YCbCr.

Run the script with the same images. Compare components of each image with each other ✍. Compare with the results obtained with the previous script ✍.

1.3. Develop a similar script but instead of converting to HSV convert to L*a*b.

Run the script with the same images. Compare components of each image with each other ✍. Compare with the results obtained with the previous script ✍.

1.4. Write a Matlab script that generates color histograms for a coloured image represented in the two color spaces, RGB and L*a*b. Compare the type of information obtained in the several color spaces

2. Variation of image spatial dimensions using or not filters with the “imzoneplate” test image

In this part, you must develop a Matlab script named “ampliaReduz.m”, which must use the “imzoneplate.m” script that is available in the UC's Moodle (obtained from the MathWorks website). Similar versions are available for Python. Before starting the work, analyze the code of the programs provided in order to understand the operations carried out.

The program to be developed “ampliaReduz.m” must allow enlarging/reducing the spatial dimensions of a “zone plate” test image. This test image should be created during the execution of the program using the “imzoneplate.m” function. It should allow enlarging/reducing by simply repeating/eliminating pixels and enlarging/reducing using Matlab's built-in function “imresize.m”. You may have the option of using an averaging or gaussian filter before enlarging/reducing. The program must display the original and processed images on the screen, as well as the respective graphs of spectral density and variation of the signal in space 📈.

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

2.1. develop your program “ampliaReduz.m” and run it with different dimensions for the zoneplate image and using different interpolation methods of the built-in function “imresize.m”. ✍️ Review the built-in “resample.m” function Run this function using the zoneplate test image. Compare and interpret the results. 📈 ✍️

3. Filtering experiences

In these experiments, you must develop a program that allows to perform different types of image filtering. The programs needs to receive the filename of the image to be filtered and the type of filter to apply. In Matlab you may use the built-in functions fspecial and imfilter to create different types of filters. In the command window of Matlab, type in “help imfilter” and “help fspecial” to understand what these functions do and how to use them. Basically the function fspecial allows to create different types of filters, whereas the function imfilter allows to apply the filter to the image.

In Python, the packages cv2 (OpenCV) or skimage have a number of modules and functions for filtering images. Some are described in the document ““intro to Jupiter notebooks.pdf” available in the Moodle section Assignment 1. Demonstrative scripts are also available in the Moodle section Assignment 2.

Criteria for Success

It is important that you get acquainted with the the two operations that need to be performed for digitising media signals, notably audio. Upon conducting this assignment you should be able to describe the effects that sampling has on the quality of the signal, discuss the ability of the human ear to perceive differences in sampling rates, the type of distortions that may arise due to the sampling process and how to minimise them. Likewise, you should have a clear understanding of the quantisation operation, the impact on the quality of the signal, discuss the severity of distortions introduced and perceived by the human ear according to the type of content and quantisation levels and explain where the distortions come from. You should also understand at a high level the dithering technique and discuss its benefits. Finally, you should understand the practical aspects of performing filtering.

Report should be delivered up to 22 March in Moodle.

Note: The template used to describe this assignment is adapted from the University of Las Vegas, Nevada, and the Transparency in Learning and Teaching (TILT) Project.