Proeskpert
Spring 2017
Data Science Assignment

Overview

The assignment will demonstrate how well you can solve problems in unknown domains, how good are your coding skills and how well you can document and share your work. As the Data Science Quiz tests your knowledge of machine learning and statistics, the practical task concentrates more on your engineering skills and is not directly related to machine learning.

The goal of the assignment is to combine 31 multispectral images into a colored image.

The points are given in the following way:

1. (25 points) The idea behind the solution;
2. (15 points) Similarity between the original image and generated image;
3. (10 points) Code quality;
4. (10 points) Documentation.

Task description

Your task is to combine images from *multispectral_images* folder into a regular colored image. The resulting image should be similar to *sponges_RGB.bmp* found in root directory. It doesn't have to be identical, but the colors should be recognizable and correspond to the ones in original image.

The approach can be implemented in a chosen programming language. Avoid using too many external libraries and tools, such as OpenCV or some specific library for combining multispectral images. Libraries for reading, displaying and saving images are welcomed though, as are other low-level libraries like numpy in Python. Using code snippets (e.g. from Stackoverflow) is allowed, but it should be mentioned in code comments.

The code should be delivered in a way that it can be run as easily as possible on Windows, Linux or Mac, including documentation for installing dependencies, packages and virtual environments. It is preferred that the code is delivered as a private git repository, for example in Bitbucket.

Short documentation should also be included in git repository with instructions for running the code and brief explanation of the idea behind the solution.

Theory

Visible spectrum is the portion of electromagnetic spectrum, which contains waves with wavelengths from around 400nm to 700nm. Regular photos are taken by looking at the whole visual spectrum. Multispectral images can be created by looking at finer portions of the spectrum and creating a grayscale image for each spectrum band. By taking multispectral images over the whole visible spectrum (or even broader spectrum) it is possible to get more detailed information about the properties of the object. Multispectral images are often used in remote sensing, but can also be used for investigating paintings, ancient papyri and classifying materials.

Regular images are often stored in RGB format, having three channels - red, green and blue. This means that 512x512 pixel color image can be viewed as a multidimensional array with dimensions 512x512x3, where element value is between 0 and 255. If for example all values in red channel are 255, and all values in other channels are 0, we get red color. If all values from all channels are 128, we get gray. If all values from all channels are 255, we get white.

More information about visible light and color models can be found at
http://lodev.org/cgtutor/color.html

Data

The multispectral images can be found in *multispectral_images* directory. The content of this directory includes full spectral resolution reflectance data from 400nm to 700nm at 10nm steps (31 bands total). Each band is stored as a 512x512 dimensional 16-bit grayscale PNG image. Image filenames are of the format *sponges_ms_01.png*, where the '01' at the end signifies that this is the first image (captured at 400nm). Thus, '02' corresponds to 410nm, and so on, until '31' for 700nm. More data about these images:
http://www1.cs.columbia.edu/CAVE/databases/multispectral/

Suggestions

If you'd like to use Python, then these following libraries could be helpful:

- *os* (*listdir*() for displaying directory content as a list of filenames)
- *scipy.misc* (*imread*() for reading image files)
- *numpy* (simplifies working with multidimensional arrays)
- *matplotlib.pyplot* (*imshow*() for displaying multidimensional arrays as images)

If you prefer to use R, you can make use of the png package to load an individual image in as a 512x512 matrix. A sample snippet to load the data as a matrix and plot it in a RStudio window is as follows:

```
library('png')
img <- readPNG('object_ms_01.png')

plot(1:512, type='n')
rasterImage(img, 1, 1, 512, 512)
```

Understanding how wavelengths can be converted to R, G, B values can be useful as these can be used for giving weights to each multispectral image. For example, the pixel values in the first picture (400nm wavelength) does not contribute to the red channel, as

light color from this spectrum band is blue. Code for finding R, G, B weights in relation to wavelengths can be found at:
http://stackoverflow.com/questions/3407942/rgb-values-of-visible-spectrum