Color and multispectral Imaging

Lab. n° 2

Objective:

The goal of this lab is to study the efficiency of some of the color constancy algorithms we have studied in class. You will also combine some of them in order to obtain better results.

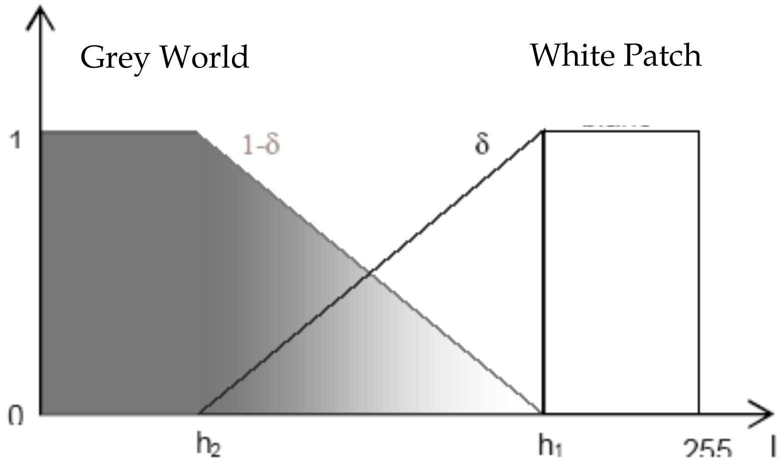
Start by copying the images in your Matlab Working Directory.

1- Write a Matlab function for each of the following algorithms:

Usage: RGB_Out = function(RGB_In)

where RGB_Out is the restored image and RGB_In is the image we want to treat.

a. Grey World
b. White patch
c. Modified White patch (h=220)
d. A progressive combination of Grey World and Modified White patch. The underlying idea consists in, locally and progressively, combining GW and MWP. Thus, the GW algorithm is applied only on dark and middle-dark areas of the image while WP algorithm is applied on brighter areas of the image. To do so, you will define two thresholds h_1 and h_2. h_1 defines the intensity threshold over which the MWP algorithm is applied. h2 defines the intensity threshold under which the GW algorithm is applied. Intensities comprised between h_1 and h_2 undergo a progressive combination of both algorithms as depicted in the figure below. The choice of the thresholds h_1 and h_2 is arbitrary and empiric.



The algorithm aims to calculate the $K_{i\ (i=R,G,B)}$ coefficients:

$$\begin{bmatrix} K_R & 0 & 0 \\ 0 & K_G & 0 \\ 0 & 0 & K_B \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} R_c \\ G_c \\ B_c \end{bmatrix}$$

We first calculate the intensity I for each pixel as follows:

$$I = \frac{R+G+B}{3}$$

If $I \geq h_1$

$$K_R = \frac{\overline{S}_r^{h_1}}{\overline{\rho}_r^{h_1}}$$

$$K_G = \frac{\overline{S}_g^{h_1}}{\overline{\rho}_g^{h_1}}$$

$$K_B = \frac{\overline{S}_b^{h_1}}{\overline{\rho}_b^{h_1}}$$

If $I \leq h_2$

$$K_R = \frac{\overline{S}_r}{\overline{\rho}_r}$$

$$K_G = \frac{\overline{S}_g}{\overline{\rho}_g}$$

$$K_B = \frac{\overline{S}_b}{\overline{\rho}_b}$$

If $h_2 \leq I \leq h_1$

$$K_R = (1-\delta)\frac{\overline{S}_r}{\overline{\rho}_r} + \delta\frac{\overline{S}_r^{h_1}}{\overline{\rho}_r^{h_1}}$$

$$K_G = (1-\delta)\frac{\overline{S}_g}{\overline{\rho}_g} + \delta\frac{\overline{S}_g^{h_1}}{\overline{\rho}_g^{h_1}}$$

$$K_B = (1-\delta)\frac{\overline{S}_b}{\overline{\rho}_b} + \delta\frac{\overline{S}_b^{h_1}}{\overline{\rho}_b^{h_1}}$$

where

$$\delta = \frac{1}{h_1 - h_2} I - \frac{h_2}{h_1 - h_2}$$

$$(\overline{\rho}_r^{h_1}, \overline{\rho}_g^{h_1}, \overline{\rho}_b^{h_1}) = \text{Average}\ (R^{h_1}, G^{h_1}, B^{h_1})$$

$$R^{h_1} \geq h_1$$

$$G^{h_1} \geq h_1$$

$$B^{h_1} \geq h_1$$

$$(\overline{S}_r^{h_1}, \overline{S}_g^{h_1}, \overline{S}_b^{h_1}) = (v^{h_1}, v^{h_1}, v^{h_1})$$

$$(\overline{\rho}_r, \overline{\rho}_g, \overline{\rho}_b) = \text{Average}\ (R\ , G\ , B\ )$$

$$(\overline{S}_r, \overline{S}_g, \overline{S}_b) = (\overline{v}, \overline{v}, \overline{v})$$

e. Single Scale Retinex and MSRCR (scale the values between 0 and 255)
f. ACE (to be tested only on the image ACE_Test.tif)

2- Test your implementations on the images in the folder Images_TP2\Color_Constancy\Test1 except ACE.

In this folder, for each set of test images there exists an image named xxx_Orig considered as the reference (acquired under canonical illuminant) and some images of the same scene acquired under non canonical illuminant.

Implement each of the following metrics in a Matlab function: MSE (global and local) and ΔE (global and local)

Usage of these functions is: metric_value = metric (xxx_Orig, RGB_Out)

<u>Note</u>: I mean with "local" that you compute the value of the metric within a ROI (Region Of Interest). Two kinds of ROI could be used: one over an area containing an edge and the other within a uniform area.

The MSE can be calculated for each couple of channels using the formula:

$$MSE = \frac{\sum_{x=1}^{m} \sum_{y=1}^{n} [(Io_{(x,y)} - Ir_{(x,y)})]^2}{m \times n}$$

where Io is the xxx_Orig image, Ir the RGB_Out image and (m,n) represent the image dimensions.
The ΔE is the Euclidean distance calculated in the Lab color space. That means you should convert your image to that space (Matlab function).

3- Provide a table that summarizes the comparison between the different algorithms using the previous metrics results. In addition, your table should contain a subjective visual criterion and running time.

<u>Note</u>: In order to obtain the running time of a Matlab program, you can place the commands *tic* in the beginning of the program and *toc* at the end.

4- Test your programs on the images contained in the folder Images_TP2\Color_Constancy\Test2.

There is no reference image xxx_Orig in this folder.

5- Discuss the efficiency and the hypothesis appropriateness of each algorithm with regards to the kind of the scenes under consideration.