Exercises chapter 2.5[·]

2.5.1. White balance and chromatic adaptation

Aim

It would be desirable that the images taken with a digital camera would be perceived as human beings view them. To achieve this purpose one of the basic aspects is the chromatic adaptation. In case of film-cameras, it is realized manually or automatically in the process of reveal. In digital cameras, it is possible to do it through software.

In this session, you are going to apply different white balance algorithms to digital images.

Fundaments

A white balance model consists of implementing a chromatic adaptation model. It is applied to compute the corresponding image taken under any light source to the image under a defined illuminant, usually D_{65}.

Thus, a chromatic adaptation model needs as input the colorimetric data (tristimulus values) of each pixel of the original image and also the colorimetric data for the original light source, or for a perfect white under it. Equivalently, the spectral reflectance of each pixel of the original image and the spectral power distribution of the light source under which the image has been taken. Usually, the colorimetric data of the light source are not available, but there are different methods to estimate them. All the methods accomplish different assumptions, more or less realistic. Following, we will show some of the easier methods:

1. Method 1.

   It assumes that the brightest object in the image is just white. Then, the color data from this object is used as original illuminant in a chromatic adaptation transform. It could be considered as the object or pixel with maximum RGB values in the image, but in your computation, please, consider the brightest object as the object with the maximum value of an achromatic channel. For this purpose, you can build an achromatic channel in the cone-response color space, as follow:

$$A=2L+M+1/20S$$

[·] Please, submit the report as pdf file and don't forget to include the Matlab code file. Both files with the format "Surname_HW_2-5-1"

2. Method 2.

Manually you can select a white object in the image and consider its color data for the application of chromatic adaptation transforms.

3. Method 3.

It is assumed that the average color data of the image is just a medium gray. Please, obtain the average color just as the mean of the RGB values of the image. Thus, the chromaticity of the average color is used along with luminance of a perfect white as input in chromatic adaptation transforms. Please, take care that only the chromaticity but not the luminance of the white will be equal to the average color of the image. To do it compute chromaticity coordinates from the average color (x_{av}, y_{av}) and then incorporate Y=100. From this data, (x_{av},y_{av},Y=100), compute tristimulus values as you know:

$$X = \frac{x}{y}Y$$

$$Z = \frac{(1-x-y)}{y}Y$$

As can be easily deduced, each of these methods has advantages and drawbacks, which you must comment on the basic of your results.

Procedure

The objective is to use the three methods explained above, along with two chromatic adaptation transforms, for the set of provided images to obtain a correct white balanced image to D_{65} illuminant (X=95.047, Y=100.00, Z=108.883).

Please, note that images “Lab2_5-1.jpg” and “Lab2_5-5.jpg” must be cut to applied different white balance to each part.

Please, use the von Kries chromatic adaptation transform and the CAT02 with each of the three methods explained above.

To obtain the color data from the image (each pixel) you can use any software you like. In case of Matlab you can use the function “imread” to do it. In any case, you need obtain the RGB values of the pixels.

From the RGB device-dependent space to tristimulus values is necessary to use a specific matrix depending on the calibration of the camera. In this lab session, we will consider the sRGB matrix. This matrix does not really fulfill the transformation from RGB to XYZ, because a calibration it is necessary to obtain the real device-dependent matrix. Nevertheless, for the purpose of this lab session, it would be clearly assumed the RGB image space as sRGB. The transformation is as follow:

- From sRGB to XYZ

The sRGB component values R_{sRGB}, G_{sRGB}, B_{sRGB} must be in the range 0 to 1. (A range of 0 to 255 can be transformed simply dividing by 255).

$$C_{linear} = \begin{cases} \dfrac{C_{sRGB}}{12.92} & C_{sRGB} \leq 0.04045 \\ \left(\dfrac{C_{sRGB} + a}{1 + a}\right)^{2.4} & C_{sRGB} > 0.04045 \end{cases}$$

$$a=0.055$$

(where C is R, G or B).

Followed by a matrix multiplication of the linear values to get XYZ:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{pmatrix} \begin{pmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{pmatrix}$$

In other text ("A Residual Modified Transformation Formula from Munsell to sRGB Color System". Shih-Wen Hsiao, Cheng-Ju Tsai) you can find the values:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119194 & 0.950227 \end{pmatrix} \begin{pmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{pmatrix}$$

You can find more information in http://en.wikipedia.org/wiki/SRGB

Please use the transformation that you prefer.

- From XYZ to sRGB

As the resulting images must be displayed the reverse transformation is needed. The first step in the calculation of sRGB tristimulus values from the CIE XYZ tristimulus values is a linear transformation.

$$\begin{pmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{pmatrix} = \begin{pmatrix} 3.2410 & -1.5374 & -0.4986 \\ -0.9692 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0570 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

The intermediate coordinates R_{linear}, G_{linear} and B_{linear} are defined in the range [0,1], which means that the initial X, Y, and Z values need to be similarly scaled (if you start with XYZ values going to 100 or so, divide them by 100 first, or apply the matrix and then scale by a constant factor to the [0,1] range).

sRGB was designed to reflect a typical real-world monitor with a gamma of 2.2, and the following formula transforms the linear values into sRGB, as follow:

$$C_{sRGB} = \begin{cases} 12.92 C_{linear} & C_{linear} \leq 0.0031308 \\ (1+a)(C_{linear})^{1/2.4} - a & C_{linear} > 0.0031308 \end{cases}$$

$$a=0.055$$

These gamma corrected values are in the range 0 to 1. If values in the range 0 to 255 are required, e.g. for video display on 8-bit graphics, the usual technique is to multiply by 255 and round to an integer.