University of Ljubljana, Faculty of Electrical Engineering
Robot Vision (RV)

March 22/23, 2016

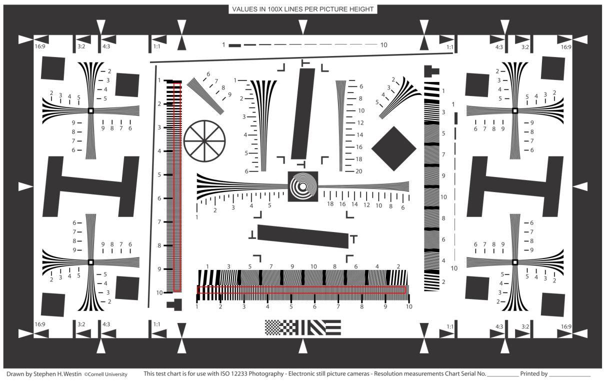
Exercise 2: Image Quality Metrics

Created by: Žiga Špiclin | http://lit.fe.uni-lj.si/RV | Homework deadline: March 29/30, 2016

Instructions

The image acquisition process, which represents a mapping from physical to image space, is influenced by several factors that may adversely impact the quality of acquired image. Among these factors are the scene illumination, sampling density, optical aberrations of the optical system, etc. The quality of the acquired image can be quantified by imaging specific calibration objects (Figure 1), based on which the image quality metrics like contrast, spatial and grayscale resolution and signal-to-noise ratio.

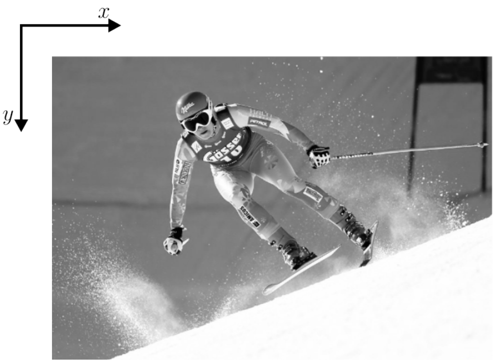
(a) ColorChecker



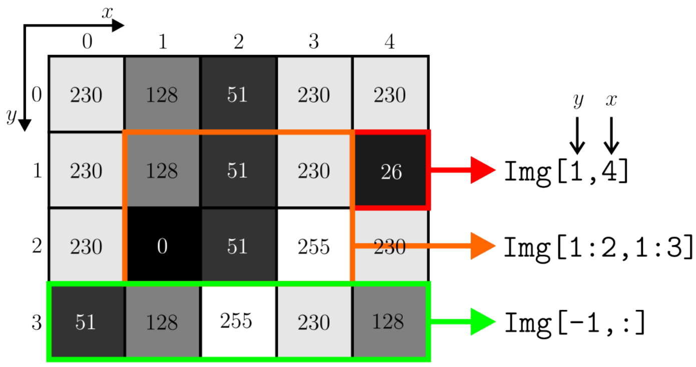
(b) ISO 12233

Figure 1: Two calibration object, which are used to quantify image quality metrics, for instance, (a) the quality of grayscale and color reproduction by the grayscale resolution, effective dynamic range and signal-to-noise ratio and (b) spatial resolution using the modulation transfer function.

Image coordinate system



Indexing pixels in the image



During this exercise you will learn how to quantify several image quality metrics and to verify the impact of external factors onto the image quality.

1. Use the digital camera on your phone to acquire two images of the ColorChecker caliber (Figure 1a), first when the lights in the lecture room are turned off and second when they are on. Ensure that your flashlight is turned off, while acquiring image. Acquire an image of ISO 12233 caliber (Figure 1b), when the lights are turned on. Transfer the acquired image onto the computer and load them into Spyder (Python) environment using function `open()` in the library package `PIL.Image`.

2. Convert the acquired image into grayscale image using equation $S = 0,299R + 0,587G + 0,114B$. To load and save the variables of type `numpy.ndarray` onto hard drive and into Spyder (Python) from a file use the respective function `tofile()` and `fromfile()`.

3. Extract the rectangular subimages from each of the acquired images of caliber ColorChecker, such that each subimage will contain one of the fields enumerated from 1 to 24. Save the subimages of type `numpy.ndarray` into a row vector of type `list` in the order of enumerated fields. For a fast and easy extraction of the subimages you can locate only the corners of a rectangle by using function `ginput()` from library package `matplotlib.pyplot`.

4. Extract two rectangular subimages from the acquired image of caliber ISO 12233 as shown on Figure 1b. Slice each of the two subimages of type `numpy.ndarray` into 10 subimages along the longest axis and save them into a row vector of type `list`. The subimages should be ordered from 1 to 10 as indicated on the Figure 1b, whereas 1 denotes the most sparse and 10 the most dense line pairs.

5. Write a function that computes contrast metric m on grayscale subimages of the caliber ISO 12233 given in `iImages`:

```
def computeContrast( iImages ):
        return oM
```

   which return a list of scalar values `oM` with the same number of elements as the number of input subimages. The minimal and maximal values f_{min} and f_{max} should be determined as the 5th in 95th percentile of all grayscale values within the subfigure using function `percentile()` in library package `numpy`. Verify that the highest contrast is obtained on the subimage with the most sparse (horizontal or vertical) line pairs.

6. Write a function that computes the effective dynamic range b_{ef} on grayscale values of the caliber ColorChecker subimages given in `iImages`:

```
def computeEffDynRange( iImages ):
        return oEDR
```

   which returns a scalar value `oEDR`. Values L_{min} and L_{max} should be computed as mean values μ of the darkest and brightest subimage, respectively. The standard deviation of noise σ_n should be estimated as the mean of standard deviations across all subimages. Verify that the return value `oEDR` is higher on the image acquired with lights in the lecture room turned on.

7. Write a function to compute the differential signal-to-noise ratio SNR_D between two arbitrary grayscale images `iImage1` and `iImage2`:

```
def computeSNR( iImage1, iImage2 ):
        return oSNR
```

   which returns a scalar value `oSNR`. Compute the SNR_D between all pairs of subimages with indices from 1 to 16 of the caliber Colorchecker and verify that the obtained values of SNR_D are larger on the image acquired with lights in the lecture room turned on.

Homework Assignments

Homework report in the form of a Python script entitled `NameSurname_Exercise2.py` should execute the requested computations and function calls and display requested figures and/or graphs. It is your responsibility to load library packages and provide supporting scripts such that the script is fully functional and that your results are reproducible. The code should execute in a block-wise manner (e.g. `#%% Assignment 1`), one block per each assignment, while the answers to questions should be written in the corresponding block in the form of a comment (e.g. `# Answer: ...`).

1. Compute the mean grayscale values μ_i of the subimages enumerated $i = 19, \ldots, 24$ in the two acquired image of caliber ColorChecker and plot these values as curves on the same graph. The graph should show the values of μ_i with respect to the enumeration i. Determine the equation that best describes the two curves and elaborate on their differences. How do these curves differ when computed for images with lights in the lecture room turn on or off?

2. Compute the modulation transfer functions (MTFs) for subimages of the caliber image ISO 12233 with horizontal and vertical line pairs. Plot the obtained MTF curves into the same graph. On each of the graphs locate the point that corresponds to value $m_{50\%}$ and use this values to determine the spatial resolution of your digital camera. Compute the spatial resolution of horizontal and vertical line pairs in line pairs per mm (lp/mm). Note that the actual caliber height is $H = 145$ mm, while the numbers (1–10) denote a 100× multiplicative factor of line pairs per picture height H, which enables to compute the *number of line pairs* per milimeter (lp/mm).

3. Determine the maximal possible value of the effective dynamic range b_{ef} of your digital camera based on the acquired images.

4. Find the pair of grayscale subimages of the ColorChecker caliber, which yields the highest value of SNR_D and list the corresponding index pair (1–24). Would it be possible to determine the obtained index pair without computing the SNR_D? Please elaborate on your answer.

5. Find those pairs of grayscale subimages of ColorChecker caliber, in which SNR_D is less than one and list the corresponding index pairs (1–24). Can you visually differentiate between such pairs of subimages (with $SNR_D < 1$) if they are displayed side-by-side? Please elaborate on your answer.