Computer Assignment 1

Due Date:
02/05/2018

1 Exercises

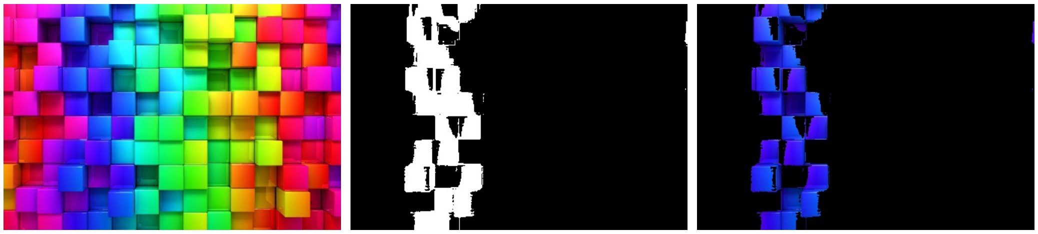
1.1 Color conversion

In class we learned about different color representations. HSL (Hue, Saturation, Lightness) and HSV (Hue, Saturation, Value) are two alternative representations of the RGB color model. Hue represents dominant color as perceived by an observer and can be used for simple color detection. Take the following steps:

- Load the ”`colors.jpg`” from sample images as a colored image. You can refer to `Example4.py` for loading a colored image.
- Use `cv2.cvtColor()` function to convert the BGR image to HSV. As an input to this function you should use `cv2.COLOR_BGR2HSV`.
- The lower and upper limits for the blue color are `[110,50,50]` and `[130,255,255]`, respectively. Use `cv2.inRange()` function to find the pixels in the HSV image that their values are in the range of blue color. Your result should be a binary mask.
- Use `cv2.bitwise_and()` function to create an image containing only the blue parts.
- Display the original image and the mask and the detected blue parts. For displaying and saving images you can refer to `Example4.py`

The threshold values for the blue color are given in this exercise. Explain how these threshold values can be obtained for different colors.

Note: You can refer to the following link for the same task done in a stream of video.
http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_colorspaces/py_colorspaces.html#converting-colorspaces



(a) Original Color image (b) Mask image (c) Segmented image

Figure 1: Resulting Figures

1.2 Histogram Operations

In this exercise we will explore the histogram equalization of gray-scale images. For quick review of the topic you can refer to the following Wikipedia page:

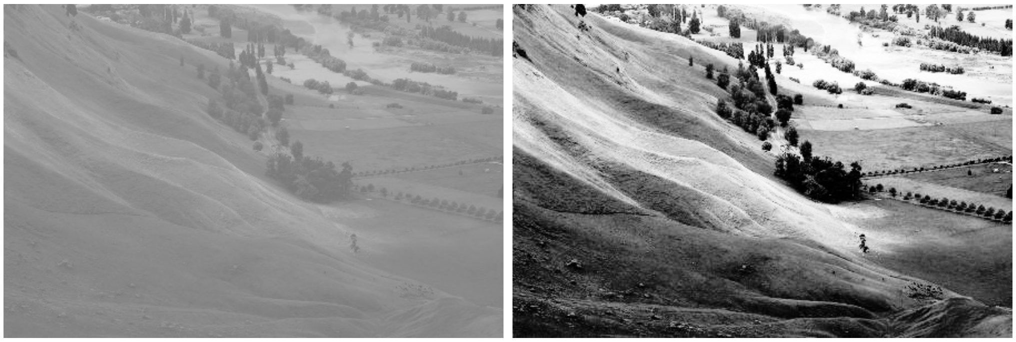
https://en.wikipedia.org/wiki/Histogram_equalization

You can also find useful sample codes in the following link from OpenCV tutorials:

http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_histograms/py_table_of_contents_histograms/py_table_of_contents_histograms.html

Take the following steps:

- Load the ”`wiki.jpg`” image from sample images as a gray-scale image using `cv2.imread()` function. Note: set the second argument of `cv2.imread()` function to zero for gray-scale.
- Calculate the histogram of the image. `numpy.histogram()` function can be used for calculation of the histogram. Refer to following link for more information: https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.histogram.html
- Calculate the cdf using the `np.cumsum()` function and proper normalization.
- Compute and apply the histogram equalizing function using the algorithm described in class.
- Display the results.



(a) Original image

(b) Histogram equalized image

Figure 2: Resulting Figures

2 Submission Format

- The submitted materials should include all your codes and your report.
- Include all files needed to run and verify your work.
- The assignment is due on 02/05/2018. Submission link will be closed at midnight and late submissions are not accepted.
- Please upload a single zip file. Name your file with your Net-ID followed by your name.