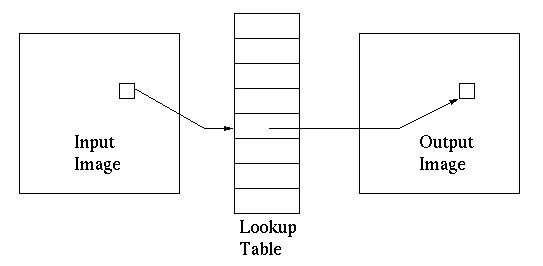
CV (Spring 2021)

Name:

Project 01. Simple Point Operations with OpenCV

Due Date: 1/22/2021, by 11:59 PM

**Read the slides on point operations.**



**Introduction:**

Point Operation are also called mapping functions where the output at (x,y) in the output image, g, depends on the point (x,y) in f, the input image.

* + Change brightness without regard to position in the image.
  + These operations are memoryless: Don’t have to remember any other points except at (x,y). This means that operations can be done in real time (the system is causal)
  + These operations, from a computation point of view, can be done in place (eventually replacing f by g) without requiring extra storage (another storage for g)
  + They work on one pixel at a time, i.e. a 1x1 neighborhood
  + They can be described either by a transformation equation or graphically by an input/output relationship.
    - Graphically the x-axis represents the input gray level, and the y-axis represents the output gray level
    - A LUT (look up table) can be used to implement the relationship
      * Entries of the LUT would be calculated off-line
      * G[x][y]= LUT[ f[x][y] ]

**Project:**

1. Apply the three following point operations on a set of images of images. You should be able to find many sets of images on the internet. You will apply the point operations on the gray level version of the image in two ways: one with the formula given and one by creating the LUT for each operation.
2. Thresholding: **V**



1. Negative image: **V**



1. Log transform: **V**



In your report, include sample of the results for each operation (the images below are just place holder, they are not the real results of these operations!)

|  |  |  |  |
| --- | --- | --- | --- |
| **Point Operation Comparison** | | | |
| Sample Image | **Operation-01** | **Operation-02** | **Operation-03** |
| Sailboat on still lake |  |  |  |
| Sailboat on still lake |  |  |  |
| Etc…. |  |  |  |
| Etc…. |  |  |  |
| Etc…. |  |  |  |
| Etc…. |  |  |  |

1. Time the two approaches (python some timing utilities such as timeit), and see if there is any difference for 1, 10, 20, 30, and 50 images. Create a word-document report with a table comparing the two approaches, similar to the following.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Point Operation Comparison** | | | | | | |
| No of images | **Operation-01** | | **Operation-02** | | **Operation-03** | |
|  | Formula | LUT | Formula | LUT | Formula | LUT |
| 1 | ***xx*** | ***xx*** | ***xx*** | ***xx*** | ***xx*** | ***xx*** |
| 10 | ***xx*** | ***xx*** | ***xx*** | ***xx*** | ***xx*** | ***xx*** |
| 20 | ***xx*** | ***xx*** | ***xx*** | ***xx*** | ***xx*** | ***xx*** |
| 30 | ***xx*** | ***xx*** | ***xx*** | ***xx*** | ***xx*** | ***xx*** |
| 50 | ***xx*** | ***xx*** | ***xx*** | ***xx*** | ***xx*** | ***xx*** |

**Grading and Submission Guide:**

* Must submit the whole project (python folder with code, image dataset, and results) zipped using 7zip tools with the name: LastName\_FirstName\_Project-01.
* For this project, put a sample of screen shots of your program run in the report.
* This is an **individual** project: The work should represent your own: that you acknowledge that have not incorporated into this project any unacknowledged material from the work of another person, including papers, words, ideas, information, computer code, data, evidence-organizing principles, or style of presentation taken from the Internet, books, periodicals, or other sources.