HOMEWORK ASSIGNMENT

NAZARBAYEV UNIVERSITY | SCHOOL OF SCIENCE AND TECHNOLOGY

PROJECT 2

In this project students will practice on their knowledge of quantization, intensity transformations. The assignment also evaluates the mathematical background and programming skills of a student.

DUE DATE

23 February 2018, Friday.

METHOD OF DELIVERY

Assignment deliverables (code, report, initial and final images) should be submitted via Moodle to the course instructor before the due date.

LEVEL OF COLLABORATION ALLOWED

Collaboration is not allowed on this assignment – each student should perform the assignment individually.

ESTIMATED TIME FOR COMPLETION

20 hours

ADDITIONAL SUPPORT

Please contact the course instructor and/or teaching assistants if you need any assistance or have any concerns about this assignment.

ASSIGNMENT DELIVERABLES

Students are expected to submit:

- MATLAB function – implementation of the task.
- Report – which describes in detail how the student approached the problem, designed a solution and implemented it.
- Associated image files generated during solution (usually placed and described within the report)

GRADING CRITERIA

60% - Black and White image task accomplishment, quality of work and final report evaluation

25% - RGB with reduced color map with 512 colors

15% - Implementation of additional dithering other than Floyd–Steinberg method

ASSIGNMENT DETAILS

In one of the last in-class exercises you were assigned to change the level of quantization of an image. In the case of two levels usually you were obtaining an image that only showed very big connected regions of black and white (BW) and it was hard to identify what was depicted on the image. There is a way to

circumvent this issue. Due to the nature of our eye, we can trick it to see more colors than it is in reality. For example, the density of black pixels on a BW image can approximate average grayscale level of the image (see Figure 1). The image on the left is grayscale image with 256 levels, and the image on right has only two levels: black and white. You see gray tones on the right, however it is only an illusion.

Figure 1. Floyd–Steinberg dithering

"Dither is an intentionally applied form of noise used to randomize quantization error, preventing large-scale patterns such as color banding in images."

One of the methods for dither and you are required to implement it in this project is Floyd–Steinberg dithering. The pseudocode of Floyd–Steinberg dithering:

```
for each y from top to bottom
   for each x from left to right
      oldpixel  := pixel[x][y]
      newpixel  := find_closest_palette_color(oldpixel)
      pixel[x][y]  := newpixel
      quant_error  := oldpixel - newpixel
      pixel[x + 1][y    ] := pixel[x + 1][y    ] + quant_error * 7 / 16
      pixel[x - 1][y + 1] := pixel[x - 1][y + 1] + quant_error * 3 / 16
      pixel[x    ][y + 1] := pixel[x    ][y + 1] + quant_error * 5 / 16
      pixel[x + 1][y + 1] := pixel[x + 1][y + 1] + quant_error * 1 / 16
```

After you accomplish grayscale dithering, to receive another 25% for the project you should implement dithering for a colored image. You should reduce color intensities to 512 level, i.e. each of your RGB channels will have only 8 intensities.

Floyd–Steinberg dithering is not the only method for accomplishing dither, there is plenty of others you should implement an additional one to get final 15% for the project. You should implement one of these dithering methods:

- Clustered dots
- Central white point
- Balanced centered point
- Diagonal ordered matrix with balanced centered points
- Dispersed dots / Bayer

You must submit a MATLAB script with a function with the following prototype:

```
robt310_project2_dither(input_file_name, output_file_name, part)
```

`input_file_name` – is an argument to indicate an input file location;

`output_file_name` – is an argument to indicate where to write output file;

`part` - is an argument to run what part to execute;