WS 2019/20	Exercises Digital Image Processing	Solution Exercise No. 5

Simple Image Processing Operations

Goal:

General

- Summary point operations (gray value- transformation functions)
- Computation of point operations
- Computation of simple Mask operations
- Boundary treatment methods

Solution for task 5.1a

Task: *What is a point operation?*

Point operations are computations referring to one specific pixel only. In principle, the value of the original pixel is replaced by the newly copmuted pixel value. Usually, all pixel values are calculated with the same computation.

Solution for task 5.1b

Task: *What point operations do you know? Give the transformation functions and the importance of these operations!*

The point operations "Range scaling of gray value" of an image and the gamma correction wwere selected for implemntation. They are described by the following operations:
Range scaling (from intervall $[a, b]$ *into intervall* $[0, 1]$*)*

$$T(r) = \frac{r-a}{b-a} \tag{1}$$

Gamma correction ($[0, 1] \rightarrow [0, c]$)

$$T(r) = c \cdot r^{\gamma} \tag{2}$$

For implemntation, the equations (1) and (2) have to be adjusted to the applied range of gray values $[0, 255]$:

Range scaling ($[a, b] \rightarrow [0, 255]$)

$$T(r) = 255 \cdot \frac{r-a}{b-a} \tag{3}$$

Gamma correction ($[0, 255] \rightarrow [0, 255]$)

$$T(r) = 255 \cdot \left(\frac{r}{255}\right)^{\gamma} \tag{4}$$

Solution for task 5.1c

Task: *Write a program that can perform the point operations from task b).*

Determination of interface:

```
// Scaling the range of gray value from [min,max] to [0,255]
void scale (GrayImage& input, GrayImage& output);
```

Implementation:

```
//
// (4.1c)   Scaling
//
void scale (GrayImage& input, GrayImage& output)
{
    int      size   = input.getSize();
    float* idata = input.getData();
    float* odata = output.getData();

    float  mini  = min(input);
    float  maxi  = max(input);

    for (int i=0; i<size; i++)
    {
        odata[i] = 255.0 * (idata[i] - mini) / (maxi - mini);
    }
}
```

Determination of interface:

```
// Gamma correction of the image with parameter <gamma>
void gamma (GrayImage& input, GrayImage& output, float gamma);
```

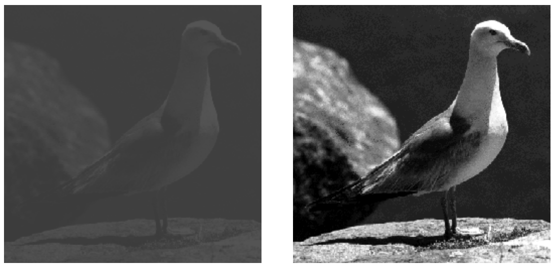
Implementation:

```
//
// (4.2b)  Gamma correction
//
void gamma (GrayImage& input, GrayImage& output, float gamma)
{
    int    size  = input.getSize();

    float* idata = input.getData();
    float* odata = output.getData();

    for (int i=0; i<size; i++)
    {
        odata[i] = 255.0 * powf(idata[i] / 255.0, gamma);
    }
}
```

The results of the implemented point operations are represented in 1 and 2.



(a) (b)

Figure 1: Scaling the gray value of an image. (a) Original image with low contrast (the gray values are in the interval [60,90]), (b) result of the "range Scaling" (the gray values now utilize the range [0,255]). The contrast of the image is much improved.

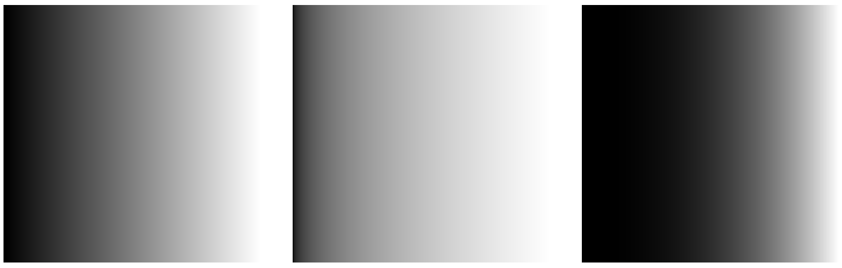


Figure 2: Gamma correction: original image of a linear gradient of gray values (a) and results of the gamma correction with $\gamma = 0.4$ (b) and $\gamma = 2.5$ (c).

Main Program:

```
int main (int argc, char** argv)
{
    /* *********************************** */
    /* Exercise 05                         */
    /* *********************************** */

    string filename;
    cout << "Filename: ";
    cin >> filename;

    GrayImage image;
    image.load(filename);
    GrayImage result(image.getWidth(), image.getHeight());

    cout << "Chose a Function" << endl;
    cout << "----------------" << endl;
    cout << "1 - Scaling" << endl;
    cout << "2 - Gamma correction" << endl;

    int input;
    cin >> input;


    switch (input)
    {
        case 1:
            scale(image, result);
            break;
        case 2:
            cout << "Gamma: " << endl;
            float g;
            cin >> g;

            gamma(image, result, g);
            break;
    }

    image.show();
    result.show();

    cout << "FINISHED.\n";
    return 0;
}
```