

Лабораторная работа 1. Гамма-коррекция серой градиентной заливки

Автор: Троцюк В.

url: <https://github.com/vladtrotsiuk/2020-misis-spring-image-processing/tree/master/lab1/>

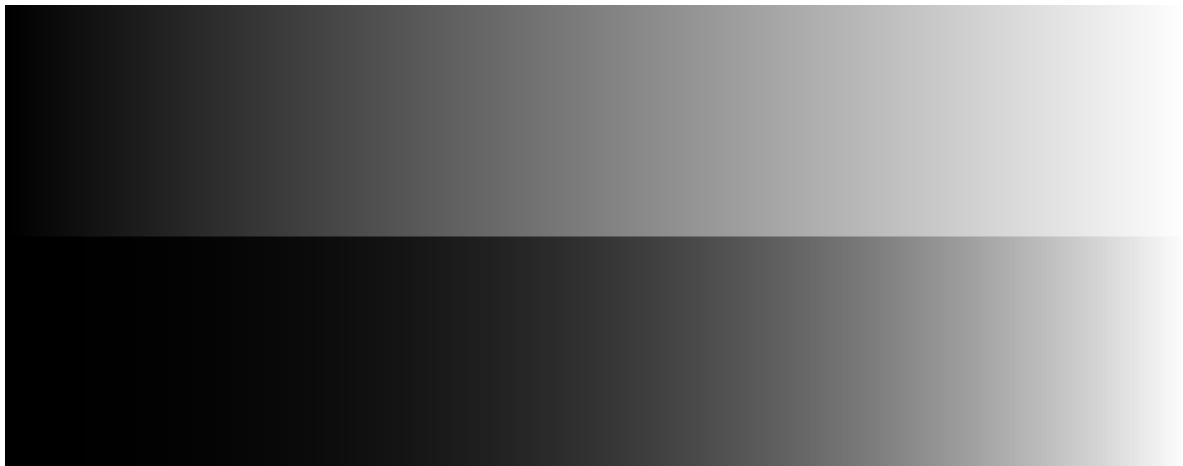
Задание

Часть 1. Нарисовать друг над другом на одном изображении \1. прямоугольник размером 768x60 пикселя с плавным изменением пикселей от черного к белому, одна градация серого занимает 3 пикселя по горизонтали. \2. изображение этого градиента после гамма-коррекции с коэффициентом из интервала 2.2-2.4

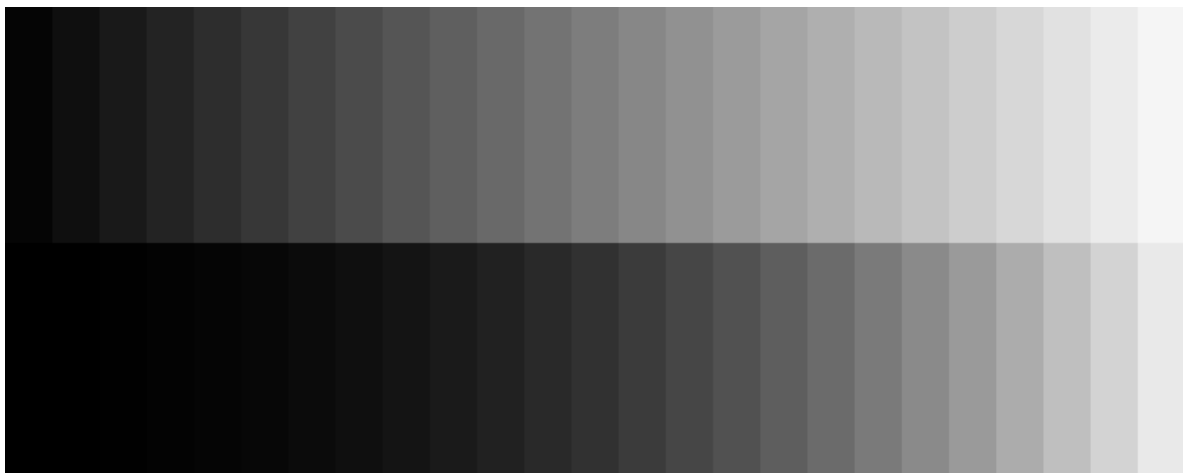
Часть 2. Нарисовать друг над другом \1. прямоугольник размером 768x60 пикселя со ступенчатым изменением пикселей от черного к белому (от 5 с шагом 10), одна градация серого занимает 30 пикселя по горизонтали. \2. изображение этого градиента после гамма-коррекции с коэффициентом из интервала 2.2-2.4

Результаты

Часть 1:



Часть 2:



Текст программы

```
#include <opencv2/opencv.hpp>

using namespace cv;

int main(int argc, char* argv[])
{
    //1
    int color1 = -1;
    Mat image1 = Mat(150, 255 * 3, CV_8UC1);
    for (int icol = 0; icol < image1.cols; icol++)
    {
        if (icol % 3 == 0)
        {
            color1++;
        }
        image1.col(icol).setTo(color1);
    }
    Mat image_2;
    Mat out_image_1;
    image1.convertTo(image_2, CV_32FC1, (float)1 / 255);
    pow(image_2, 2.3, image_2);
    image_2.convertTo(out_image_1, CV_8UC1, 255);

    image1.push_back(out_image_1);
    imshow("image", image1);
    imwrite("../res_part1.png", image1);

    std::cout << "Press any button!";
    waitKey(0);

    //2
    Mat out_image(300, 25 * 30, CV_8UC1);
    int color = -5;
    Mat roi1 = out_image(Rect(0, 0, out_image.cols, out_image.rows / 2));
    Mat roi2 = out_image(Rect(0, 150, out_image.cols, out_image.rows / 2));

    for (int icol = 0; icol < roi1.cols; icol++)
    {
        if (icol % 30 == 0)
        {
            color += 10;
        }
        roi1.col(icol).setTo(color);
    }

    roi1.convertTo(roi2, CV_32FC1, (float)1 / 255);
    Mat roi3 = out_image(Rect(0, 150, out_image.cols, out_image.rows / 2));

    pow(roi2, 2.3, roi2);
    roi2.convertTo(roi3, CV_8UC1, 255);

    imshow("image", out_image);
    imwrite("../res_part2.png", out_image);
}
```

```
waitKey(0);  
return 0;  
}
```