lab01.report.md 2/14/2022

## Работа 1. Исследование гамма-коррекции

автор: Измайлов Л.С. дата: 2022-02-14T15:03:32

url: https://github.com/lzlaster/izmailov\_l\_s/tree/main/prj.labs/lab01

## Задание

- 1. Сгенерировать серое тестовое изображение \$I\_1\$ в виде прямоугольника размером 768x60 пикселя с плавным изменение пикселей от черного к белому, одна градация серого занимает 3 пикселя по горизонтали.
- 2. Применить к изображению \$I\_1\$ гамма-коррекцию с коэффициентом из интервала 2.2-2.4 и получить изображение \$G\_1\$ при помощи функци роw.
- 3. Применить к изображению \$I\_1\$ гамма-коррекцию с коэффициентом из интервала 2.2-2.4 и получить изображение \$G\_2\$ при помощи прямого обращения к пикселям.
- 4. Показать визуализацию результатов в виде одного изображения (сверху вниз \$I\_1\$, \$G\_1\$, \$G\_2\$).
- 5. Сделать замер времени обработки изображений в п.2 и п.3, результаты отфиксировать в отчете.

## Результаты

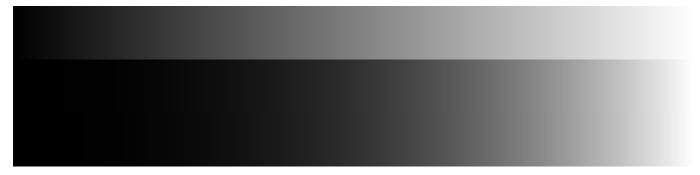


Рис. 1. Результаты работы программы (сверху вниз \$I\_1\$, \$G\_1\$, \$G\_2\$)

## Текст программы

lab01.report.md 2/14/2022

```
G_1.convertTo(G_1, CV_8UC1, 255.0 / 1.0);
    tm1.stop();
    cv::Mat G_2;
    G_2 = I_1.clone();
    tm2.start();
    for (int i = 0; i < G_2.rows; i++)
        for (int j = 0; j < G_2.cols; j++)
            G_2.at<uint8_t>(i, j) = std::pow(G_2.at<uint8_t>(i, j) / 255.0, 2.4) *
255;
    tm2.stop();
    cv::Mat img;
    img = 0;
    cv::Mat mArr[] = { I_1, G_1, G_2 };
    cv::vconcat(mArr, 3, img);
    std::cout << "G_1 Total time: " << tm1.getAvgTimeSec() << std::endl;</pre>
    std::cout << "G_2 Total time: " << tm2.getAvgTimeSec() << std::endl;</pre>
    cv::imwrite("lab01.png", img);
    cv::waitKey(∅);
}
```