

Работа 1. Исследование гамма-коррекции

автор: Кузнецов Д.А. дата: 2022-03-07T12:52:42

Задание

1. Сгенерировать серое тестовое изображение I_1 в виде прямоугольника размером 768x60 пикселя с плавным изменением 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 функции `row`.
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)

Время выполнения п.2: 0.0072267

Время выполнения п.3: 0.0078592

Текст программы

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

int main() {
    cv::Mat I_1(60, 768, CV_8UC1);
    // draw dummy image
    I_1 = 0;

    cv::imshow("dmmaopd", I_1);

    cv::TickMeter timer;

    //making single channel picture with gradient
    for (int i = 0; i < I_1.rows; i++) {
        for (int j = 0; j < I_1.cols; j++) {
```

```

        I_1.at<uchar>(i, j) = j / 3;
    }
}

cv::Mat G_1 = I_1.clone();

timer.start();

G_1.convertTo(G_1, CV_32FC1, 1/255.);

cv::pow(G_1, 2.2, G_1);

G_1.convertTo(G_1, CV_8UC1, 255);

timer.stop();
std::cout << "G_1 timer = " << timer.getTimeSec() << std::endl;
timer.reset();

cv::Mat G_2 = I_1.clone();

timer.start();

for (int i = 0; i < G_2.rows; i++)
    for (int j = 0; j < G_2.cols; j++) {
        G_2.at<uchar>(i, j) = std::pow(G_2.at<uchar>(i, j) / 255., 2.2) * 255;
    }

timer.stop();
std::cout << "G_2 timer = " << timer.getTimeSec() << std::endl;

cv::Mat resoult;
cv::vconcat(I_1, G_1, resoult);
cv::vconcat(resoult, G_2, resoult);

cv::imshow("lab01.png", resoult);
cv::imwrite("lab01.png", resoult);

cv::waitKey(0);
}

```