

Работа 1. Исследование гамма-коррекции

автор: Мочалов Л.В. дата: 2022-02-17T22:52:08

url: https://github.com/RatGeneral/CV_Image/tree/main/prj.labs/lab01

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 затраченное на обработку G_1 , G_2 :

0.0058539

0.0061931

Текст программы

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

int main() {

    cv::Mat I_1(60, 768, CV_8UC1);
    int color = 0;
    for (size_t i = 0; i < I_1.rows; i++)
    {
        for (size_t j = 0; j < I_1.cols; j += 3)
        {
            I_1.at<uint8_t>(i, j) = color;
            I_1.at<uint8_t>(i, j + 1) = color;
```

```
        I_1.at<uint8_t>(i, j + 2) = color;
        color++;
    }
}

//cv::imwrite("lab01.png", I_1);

cv::Mat G_1(60, 768, CV_8UC1);

cv::TickMeter time;
time.start();

I_1.convertTo(G_1, CV_32FC1, 1.0 / 255);
pow(G_1, 2.2, G_1);
G_1.convertTo(G_1, CV_8UC1, 255);

time.stop();
std::cout << "i.2 = " << time.getTimeSec() << std::endl;

time.reset();

//cv::imwrite("lab011.png", G_1);

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

time.start();
for (size_t i = 0; i < G_2.rows; i++)
{
    for (size_t j = 0; j < G_2.cols; j += 3)
    {
        double col = std::pow(G_2.at<uint8_t>(i, j) / 255.0, 2.2) * 255;

        G_2.at<uint8_t>(i, j) = col;
        G_2.at<uint8_t>(i, j + 1) = col;
        G_2.at<uint8_t>(i, j + 2) = col;
    }
}
time.stop();
std::cout << "i.3 = " << time.getTimeSec() << std::endl;

//cv::imwrite("lab012.png", G_2);

cv::Mat images[] = { I_1, G_1, G_2 };
cv::Mat all_images;
cv::vconcat(images, 3, all_images);

cv::imwrite("lab01.png", all_images);

cv::waitKey(0);
}
```