

Работа 1. Исследование гамма-коррекции

автор: Егубова П.А.

дата: 2021-03-01T17:20:16

url: https://mysvn.ru/egubova_p_a/egubova_p_a/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 `pow`.
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

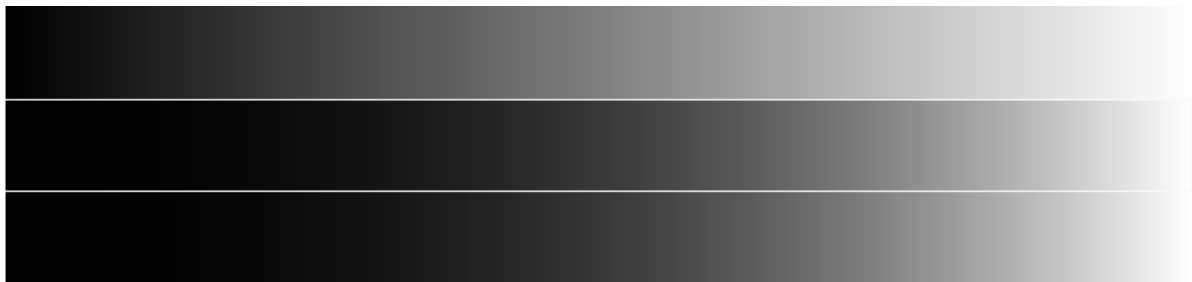


Рис. 1. Результаты работы программы (сверху вниз I_1, G_1, G_2)

Время обработки изображений

Метод 1 - 0.0005723 сек. Метод 2 - 0.0007677 сек.

Текст программы

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

using namespace cv;
using namespace std;

double const GAMMA = 2.2;

int main() {
    Mat img(180, 768, CV_8UC1);
    img = 0;
    Rect2d rc = { 0, 0, 768, 60 };
    int color = -1;
    for (int col = 0; col < 768; col++) {
        for (int row = 0; row < 60; row++) {
```

```

        img.at<uchar>(row, col) = uchar(color);
    }
    if (col % 3 == 0) {
        color++;
    }
}
rc.y += rc.height;

cv::TickMeter time;
Mat gamma1_src = img(Range(0, 60), Range(0, 768)).clone();
MatId gamma1_src_double;
gamma1_src.convertTo(gamma1_src_double, CV_64F, 1.0f / 255.0f);
MatId gamma1_dst_double;
time.start();
cv::pow(gamma1_src_double, GAMMA, gamma1_dst_double);
time.stop();
MatIb gamma_dst;
gamma1_dst_double.convertTo(gamma_dst, CV_8U, 255.0f, 0.5);
std::cout << "Method 1: " << time.getTimeSec() << std::endl;
gamma_dst.copyTo(img(rc));
rectangle(img, rc, { 250 }, 1);
rc.y += rc.height;

Mat gamma2_src = img(Range(0, 60), Range(0, 768)).clone();
Mat gamma2_dst = img(Range(0, 60), Range(0, 768)).clone();
Mat lut(1, 256, CV_8UC1);
uchar* p = lut.ptr();
time.start();
for (int i = 0; i < 256; i++) {
    p[i] = saturate_cast<uchar>(pow(i / 255.0, GAMMA) * 255.0);
}
LUT(gamma2_src, lut, gamma2_dst);
time.stop();
std::cout << "Method 2: " << time.getTimeSec() << std::endl;
gamma2_dst.copyTo(img(rc));
imwrite("lab01.png", img);
}

```