

Работа 1. Исследование гамма-коррекции

автор: Панин Г.И. дата: 2022-02-17T12:25:11

url: <https://github.com/ssimpletonn/panin-g-i/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)

Замер времени

п.2 - 13ms п.3 - 6ms

Текст программы

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

int main() {
    double gamma = 2.3;
    cv::Mat img(180, 768, CV_8UC1);
    //draw dummy image
    img = 0;
    cv::Rect2d rc = {0, 0, 768, 60 };
    cv::rectangle(img, rc, { 100 }, 1);
    rc.y += rc.height;
    cv::rectangle(img, rc, { 250 }, 1);
    rc.y += rc.height;
```

```
cv::rectangle(img, rc, { 150 }, 1);
//save result

//От черного к белому
for(int i = 0; i < 768; i++){
    for(int k = 0; k < 60; k++){
        img.at<uchar>(k,i) = i/3;
    }
}

//Гамма-коррекция pow
unsigned int start_time = clock();
rc = {0, 0, 768, 60};
cv::Rect2d rc1 = {0, 60, 768, 60 };
img.convertTo(img, CV_64F);
cv::pow(img(rc)/255, gamma, img(rc1));
img(rc1) *= 255;
img.convertTo(img, CV_8UC1);
unsigned int end_time = clock();
unsigned int td1 = end_time - start_time;

//Гамма-коррекция(прямое обращение)
start_time = clock();
for(int i = 0; i < 768; i++){
    for(int k = 0; k < 60; k++){
        img.at<uchar>(k + 120,i) = cv::saturate_cast<uchar>(cv::pow(img.at<uchar>
(k,i)/255., gamma) * 255);
    }
}
end_time = clock();
unsigned int td2 = end_time - start_time;

std::cout << "1 method: " << td1 << "ms" << '\n' << "2 method: " << td2 << "ms";
cv::imshow("img", img );

cv::waitKey(0);

cv::imwrite("lab01.png", img);
}
```