Работа 3. Яркостные преобразования

автор: Лоев В.А.

дата: 5.03.2021

https://mysvn.ru/LOEV_V_A/loev_v_a/prj.labs/lab_3/

Задание

1. В качестве тестового использовать изображение data/cross_0256x0256.png
2. Сгенерировать нетривиальную новую функцию преобразования яркости (не стоит использовать линейную функцию, гамму, случайная).
3. Сгенерировать визуализацию функцию преобразования яркости в виде изображения размером 512x512, черные точки а белом фоне.
4. Преобразовать пиксели grayscale версии тестового изображения при помощи LUT для сгенерированной функции преобразования.
5. Преобразовать пиксели каждого канала тестового изображения при помощи LUT для сгенерированной функции преобразования.
6. Результаты сохранить для вставки в отчет.

Результаты

Рис. 1. Исходное тестовое изображение

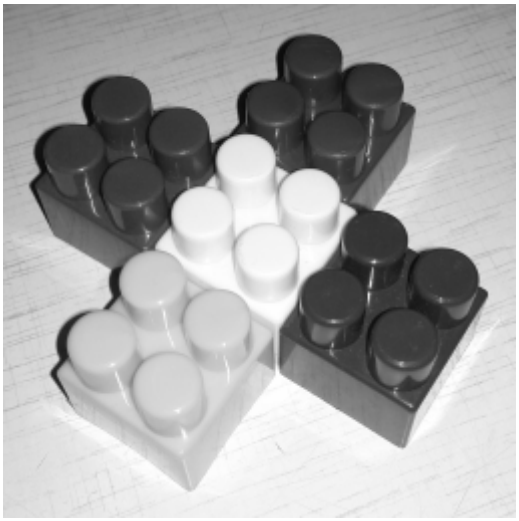

Рис. 2. Тестовое изображение greyscale

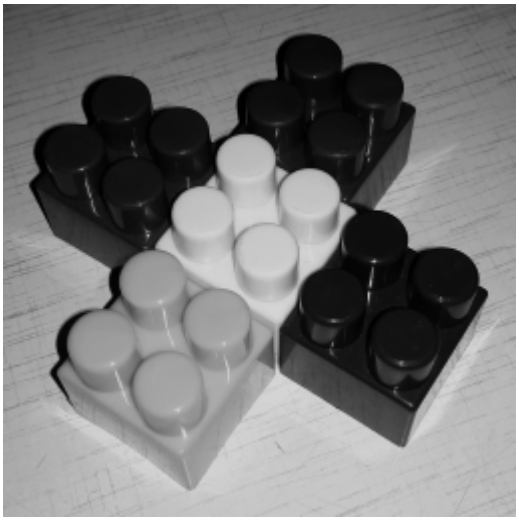

Рис. 3. Результат применения функции преобразования яркости для greyscale

Рис. 4. Результат применения функции преобразования яркости для каналов

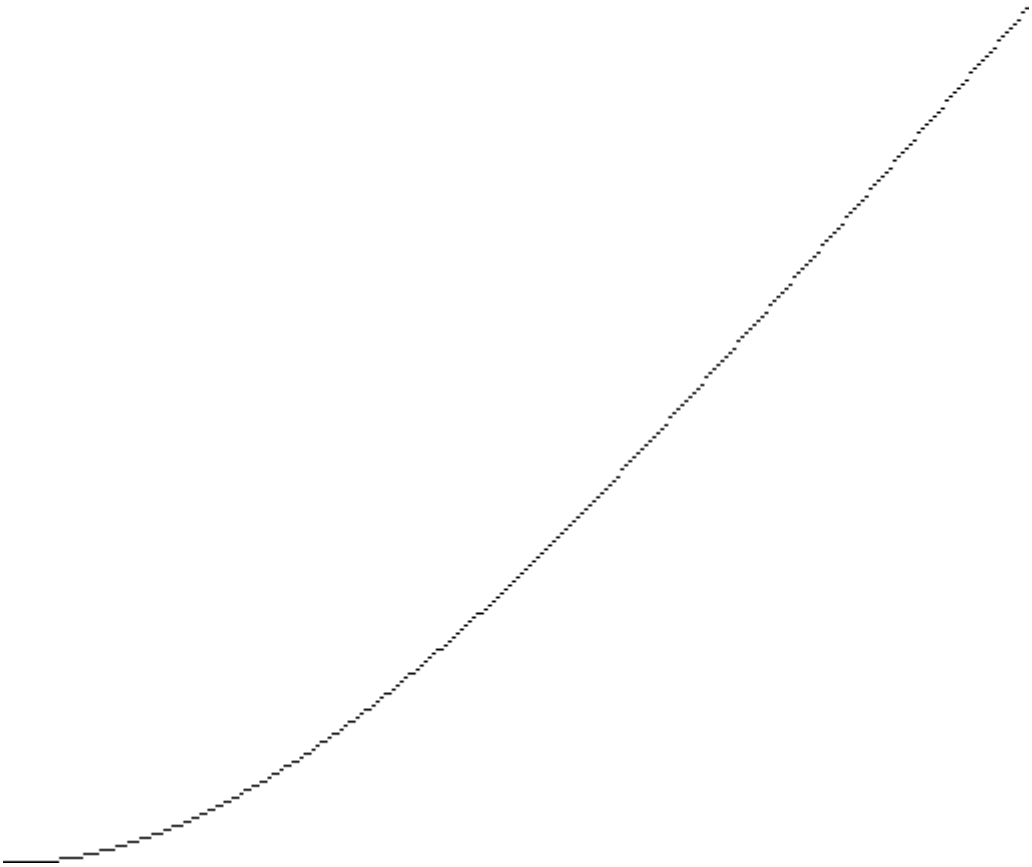

Рис. 5. Визуализация функции яркостного преобразования

Текст программы

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

#define PI 3.14159265


using namespace cv;
using namespace std;


int brightnessFunc(int x) {

    double v = double(x) / 255;
    return -3.3 * log10(v + 1) * cos(v + 0.5 * PI) * 255;
}


int main() {

    //load source image
    string image_path = samples::findFile("cross_0256x0256.png");
    Mat original_img = imread(image_path, IMREAD_COLOR);
    imwrite("lab03_rgb.png", original_img);
    imshow("original_img", original_img);

    //convert image to greyscale
```

```
    Mat grey_img;
    cv::cvtColor(original_img, grey_img, cv::COLOR_BGR2GRAY);
    imwrite("lab03_gre.png", grey_img);
    imshow("grey_img", grey_img);

    //make look-up-table
    Mat look_up_table(1, 256, CV_8U);
    uchar* p = look_up_table.ptr();

    for (int i = 0; i < 256; ++i)
        p[i] = brightnessFunc(i);

    imshow("Look_up_table", look_up_table);

    //LUT for greyscale
    Mat grey_lut_img;
    LUT(grey_img, look_up_table, grey_lut_img);
    imwrite("lab03_gre_res.png", grey_lut_img);
    imshow("gre_res", grey_lut_img);

    //LUT for original
    Mat rgb_lut_img;
    LUT(original_img, look_up_table, rgb_lut_img);
    imwrite("lab03_rgb_res.png", rgb_lut_img);
    imshow("rgb_res", rgb_lut_img);

    //Visualize brightness function
    Mat brightness_func = 255 * Mat::ones(512, 512, CV_8UC1);

    for (int i = 0; i < 256; ++i) {
        brightness_func.at<uchar>(511 - 2 * brightnessFunc(i), 2 * i) = 0;
        brightness_func.at<uchar>(511 - 2 * brightnessFunc(i), 2 * i + 1) = 0;
    }

    imwrite("lab03_viz_func.png", brightness_func);
    imshow("BrightnessFunc", brightness_func);

    waitKey(0);
    return 0;
}
```