

Работа 3. Яркостные преобразования

автор: Юркин П.О.

дата: 05.03.2021

https://mysvn.ru/Maxim_Kaverin/anime/yurkin_p_o/prj.labs/lab03

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

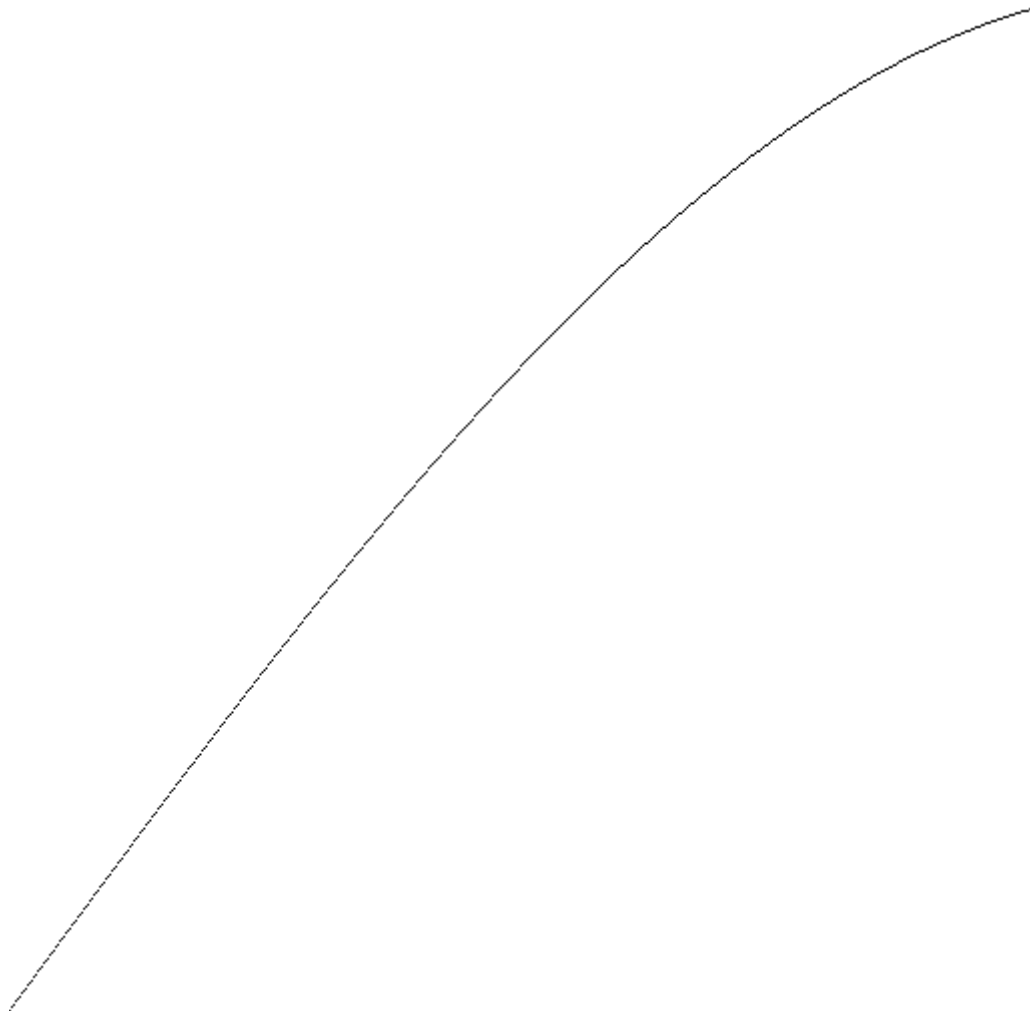


Рис. 5. Визуализация функции яркостного преобразования

Текст программы

```
#include <opencv2/opencv.hpp>
using namespace cv;

//### Задание
//1. В качестве тестового использовать изображение data / cross_0256x0256.png

float F(const float& x, const int& imageSize) {
    return sin((CV_PI * 1.3 * x / 3) / (imageSize - 1)) * (imageSize - 1);
}

Mat Plot(){
    Mat output(512, 512, CV_8U, 255);
    for (int i = 0; i < 512; ++i){
        output.at<uchar>(511 - F(i, 512), i) = 0;
    }
    return output;
}

Mat MakeBright(const Mat& img) {
    Mat output;
    std::vector<uchar> lookupTable(256);
    for (int i = 0; i < 256; ++i) {
        lookupTable[i] = F(i, 256);
    }
    LUT(img, lookupTable, output);
}
```

```
        return output;
    }

    int main() {
        Mat img = imread("data_cross_0256x0256.png");
        Mat brite_img = MakeBright(img);
        Mat gray_img;
        cvtColor(img, gray_img, COLOR_BGR2GRAY);
        Mat brite_gray_img = MakeBright(gray_img);
        Mat plot = Plot();

        imwrite("lab03_rgb.png", img);
        imwrite("lab03_gre.png", gray_img);
        imwrite("lab03_rgb_res.png", brite_img);
        imwrite("lab03_gre_res.png", brite_gray_img);
        imwrite("lab03_viz_func.png", plot);
        waitKey(0);
        return 0;
    }
}
```