

Работа 3. Яркостные преобразования изображений

автор: Мисютин В. А. дата: 2022-02-26T21:18:36

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

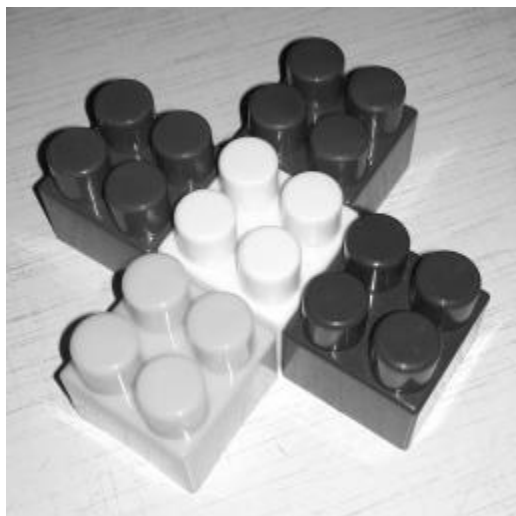


Рис. 2. Тестовое изображение greyscale



Рис. 3. Результат применения функции преобразования яркости для greyscale

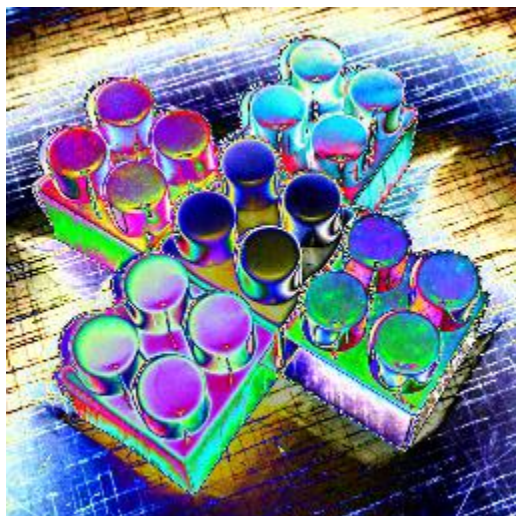


Рис. 4. Результат применения функции преобразования яркости для каналов

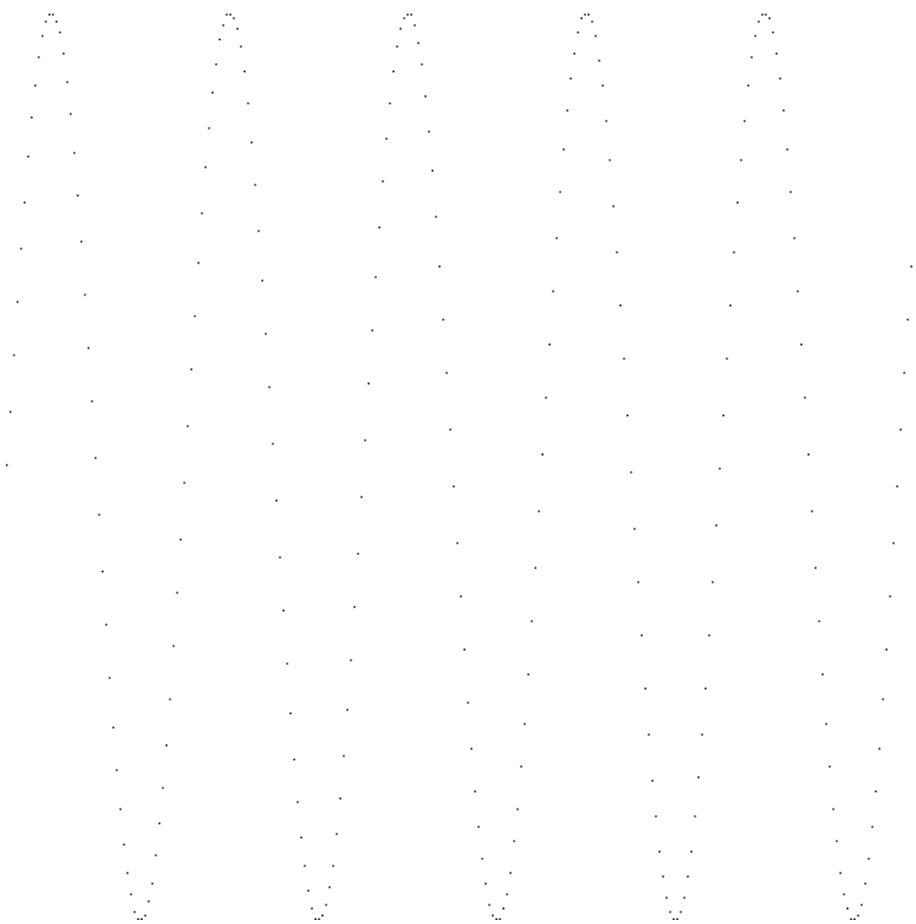


Рис. 5. Визуализация функции яркостного преобразования

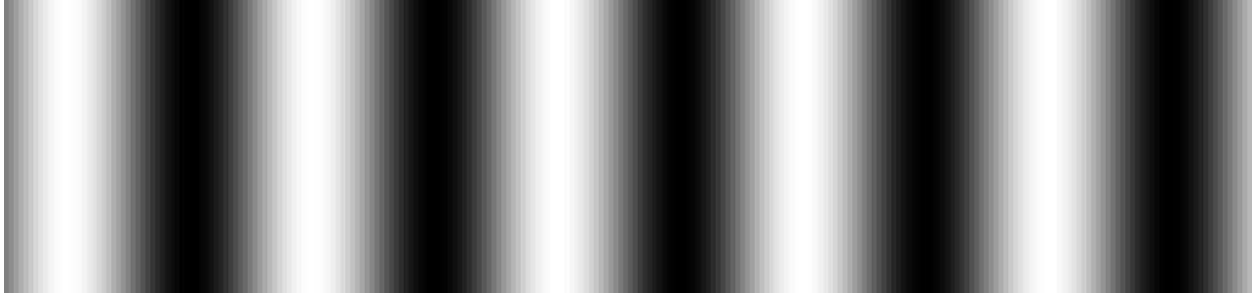


Рис. 6. Проверка функции

Текст программы

```
//  
// Created by dxmwrld on 24.02.2022.  
//  
  
#include <opencv2/opencv.hpp>  
#include <cmath>  
  
int main() {  
    cv::Mat src_img = cv::imread("../data/cross_0256x0256.png");  
    if (src_img.empty())  
        std::cout << "Could not find image" << std::endl;  
    cv::Mat m(180, 768, CV_8UC1);  
    m = 0;  
    for (ptrdiff_t y{0}; y < 180; ++y)  
        for (ptrdiff_t x{0}; x < 768; ++x)  
            m.at<uchar>(y, x) = x / 3;  
  
    cv::imwrite("Lab3_rgb.png", src_img);  
    cv::Mat GreyScale;  
    cv::cvtColor(src_img, GreyScale, cv::COLOR_BGR2GRAY);  
  
    cv::imwrite("Lab3_grey.png", GreyScale);  
    cv::Mat LookUpTable(1, 256, CV_8UC1);  
    for (ptrdiff_t i{0}; i < 256; ++i)  
        LookUpTable.at<uchar>(0, i) = static_cast<uint32_t>(128.0f * (sinf(i / 8.0f)  
            + 1.0f));  
    cv::Mat ResultForInitial, ResultForGreyscale, ResultForChecker;  
    cv::LUT(m, LookUpTable, ResultForChecker);  
    cv::LUT(src_img, LookUpTable, ResultForInitial);  
    cv::LUT(GreyScale, LookUpTable, ResultForGreyscale);  
  
    cv::imwrite("Lab3_grey_res.png", ResultForGreyscale);  
  
    cv::imwrite("Lab3_rgb_res.png", ResultForInitial);  
  
    cv::imwrite("Lab3_check.png", ResultForChecker);
```

```
cv::Mat FunctionVisualization(512, 512, CV_8UC1, 255);  
for (ptrdiff_t i{0}; i < 512; i += 2)  
FunctionVisualization.at<uchar>(512 - 2 * LookUpTable.at<uchar>(0, i / 2) -  
1, i) = 0;  
cv::imwrite("Lab3_visual.png", FunctionVisualization);  
}
```