Работа 1. Исследование гамма-коррекции

автор: Плохотнюк А. Д. дата: 2022-02-22T11:16:20

url: https://github.com/Gorgeousanya/ImageProcessing

Задание

1. Сгенерировать серое тестовое изображение I_1 в виде прямоугольника размером 768x60 пикселя с плавным изменение 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 функци 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 в мс, затраченное на гамма-коррекцию с помощью pow: 0.341727

Время в мс, затраченное на гамма-коррекцию с помощью прямого обращения к пикселям : 1.67781

Текст программы

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

int main() {
  cv::Mat img1(60, 768, CV_8UC1);
  cv::Vec3b val;
  for(int y = 0; y < 60; y++){
      val[0]=0;
      val[1]=0;
      val[2]=0;
      for(int x = 0; x < 768; x++) {
          img1.at<cv::Vec3b>(y, x) = val;
          val[0] += 1;
```

```
            val[1] += 1;
            val[2] += 1;
        }
    }

    cv::Mat img2;
    img1.convertTo(img2, CV_64F, 1.0 / 255.0);
    float gamma = 2.2;
    double t1 = (double)cv::getTickCount();
    cv::pow(img2, gamma, img2);
    double t2 = (double)cv::getTickCount();
    std::cout << "Время в мс, затраченное на гамма-коррекцию с помощью pow:
" <<1000 * ((t2 - t1) / cv::getTickFrequency()) << '\n';
    img2.convertTo(img2, CV_8UC1, 255., 0);

    cv::Mat img3;
    img1.convertTo(img3, CV_64F, 1.0);
    t1 = (double)cv::getTickCount();
    for(int y = 0; y < 60; y++){
        for(int x = 0; x < 768; x++) {
            img3.at<double>(y, x) = cv::pow(img3.at<double>(y, x)/255,
gamma)*255.0;
        }
    }
    t2 = (double)cv::getTickCount();
    std::cout << "Время в мс, затраченное на гамма-коррекцию с помощью
прямого обращения к пикселям : " <<1000 * ((t2 - t1) /
cv::getTickFrequency()) << '\n';
    img3.convertTo(img3, CV_8UC1, 1.0 , 0);

    // save result
    cv::Mat img(0, 768, CV_8UC1);
    img.push_back(img1);
    img.push_back(img2);
    img.push_back(img3);
    cv::imwrite("I_1.png", img1);
    cv::imwrite("G_1.png", img2);
    cv::imwrite("G_2.png", img3);

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

    cv::waitKey(0);
}
```